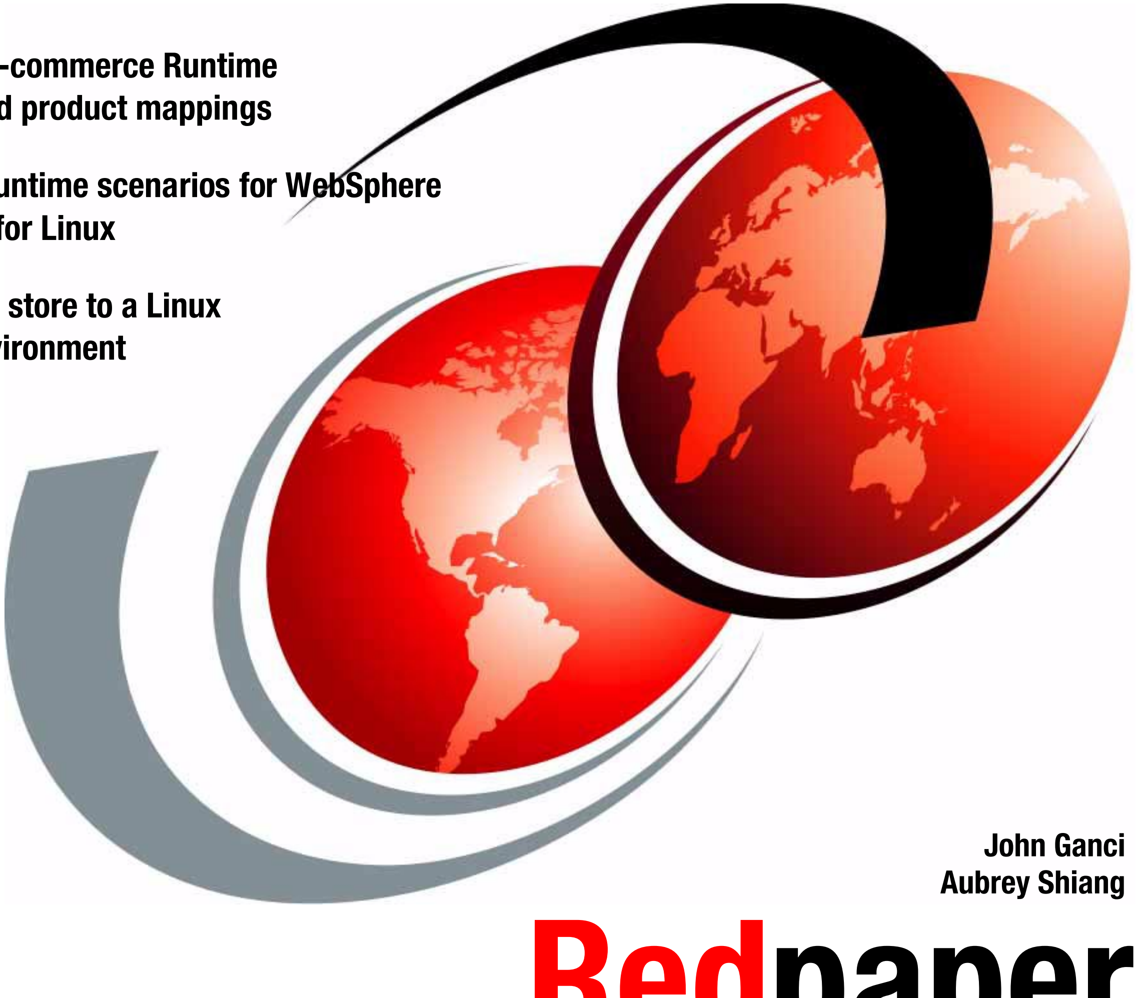
WebSphere® software

IBM

WebSphere Commerce V5.4 for Linux

Infrastructure and Deployment Patterns



- Review of e-commerce Runtime patterns and product mappings
- Advanced runtime scenarios for WebSphere Commerce for Linux
- Deploying a store to a Linux runtime environment

John Ganci
Aubrey Shiang

Redpaper

ibm.com/redbooks

International Technical Support Organization

WebSphere Commerce V5.4 for Linux Infrastructure and Deployment Patterns

November 2002

Note: Before using this information and the product it supports, read the information in "Notices" on page ix.

First Edition (November 2002)

This edition applies to IBM WebSphere Commerce V5.4, Business Edition for Linux for use with the Red Hat Linux 7.2 operating system.

Contents

Notices ix
Trademarks x

Preface xi
The team that wrote this Redpaper xi
Become a published author xii
Comments welcome xiii

Chapter 1. Introduction 1
1.1 IBM WebSphere Commerce V5.4 for Linux 2
1.1.1 WebSphere Commerce V5.4 product editions 2
1.1.2 WebSphere Commerce Studio V5.4 product offerings 3
1.1.3 Finding WebSphere Commerce information 3
1.2 Redpaper objectives 5
1.3 Structure of the Redpaper 7

Chapter 2. Working with the e-Marketplace patterns 9
2.1 e-Marketplaces 10
2.1.1 Trading Exchange composite pattern 10
2.1.2 Sell-Side Hub composite pattern 11
2.1.3 Buy-Side Hub composite pattern 12
2.2 Selecting the Application pattern 13
2.2.1 Application pattern 1: e-commerce 14
2.2.2 Application pattern 2: Self-Service 15
2.2.3 Application pattern 3: Collaboration 15
2.2.4 Application pattern 4: Batch data exchange 15
2.2.5 Application pattern 5: Extended Enterprise 16
2.3 Selecting the Runtime patterns 17
2.3.1 Runtime patterns 18
2.3.2 Runtime node definitions 23
2.4 Selecting the Product mappings 29
2.5 Summary 30

Chapter 3. Implementing a single-tier runtime 31
3.1 Planning 33
3.1.1 Hardware and software prerequisites 33
3.1.2 Hardware used in the ITSO test environment 33
3.1.3 Software used in the ITSO test environment 34
3.2 Red Hat Linux installation 36

3.2.1 Where to find more information on Linux 37
3.2.2 Linux kernel. 37
3.2.3 File systems 37
3.2.4 Linux software packages. 39
3.3 DB2 Server installation 39
3.3.1 Pre-installation for DB2 40
3.3.2 Install the DB2 Server 42
3.3.3 Verify the DB2 Server installation 46
3.3.4 Install DB2 UDB V7 FixPak 7 49
3.3.5 Configure the DB2 Server 50
3.3.6 Create the WebSphere Application Server repository database. . . . 54
3.3.7 Stop DB2. 56
3.3.8 Start DB2. 56
3.4 IBM HTTP Server and WebSphere plugin installation 57
3.4.1 Install the IBM HTTP Server 58
3.4.2 Install the WebSphere plugin for the IBM HTTP Server 59
3.4.3 Install the WebSphere Application Server V4 FixPak 4 61
3.4.4 Cumulative WebSphere plugin e-fix (PQ63508) 61
3.4.5 Configure the IBM HTTP Server 62
3.4.6 Restart, stop and start IBM HTTP Server 67
3.4.7 Verify the IBM HTTP Server 67
3.5 WebSphere Application Server installation 69
3.5.1 Pre-installation for WebSphere Application Server 70
3.5.2 Install the WebSphere Application Server. 71
3.5.3 Install WebSphere Application Server V4 FixPak 4 73
3.5.4 Install WebSphere Application Server V4.0.4 e-Fixes 75
3.5.5 Verify the WebSphere Application Server installation. 76
3.5.6 Configure the WebSphere Application Server 77
3.5.7 Verify the WebSphere Application Server. 79
3.6 WebSphere Commerce installation. 83
3.7 WebSphere Commerce Payments installation 84
3.7.1 Pre-installation for WebSphere Payment Manager V3.1.2 85
3.7.2 Install WebSphere Payment Manager V3.1.2 87
3.7.3 Install WebSphere Commerce Payments V3.1.3 FixPak 88
3.8 WebSphere Commerce post-install configuration 89
3.8.1 Execute post install script (wcb.sh) 90
3.8.2 Post wcb.sh configuration 92
3.8.3 Start WebSphere Administrative Server (non-root wasuser) 94
3.8.4 Start WebSphere Administrative Console (non-root wasuser) 95
3.8.5 WebSphere Commerce FixPak V5.4.0.3 installation 95
3.8.6 Verify WebSphere Application Server non-root user 97
3.8.7 Modify was.deployed.EJB.xml (workaround) 98
3.8.8 Update setenv.sh (workaround) 99

3.8.9 Change file ownership and permissions (workaround) 99
3.9 WebSphere Commerce instance creation. 100
3.9.1 Prerequisites for WebSphere Commerce instance creation 101
3.9.2 Start the Configuration Manager Server 101
3.9.3 Start the Configuration Manager (client) 102
3.9.4 Create an instance with the Instance Creation Wizard 102
3.9.5 Verify the WebSphere Commerce instance creation 104
3.10 WebSphere Commerce post-instance config 105
3.10.1 Regenerate the Web server plugin 105
3.10.2 Stop/start the WebSphere Commerce instance 106
3.10.3 Compile the WebSphere Commerce tools JSPs (WCS Tools). . . 107
3.10.4 Rebind data source (workaround). 109
3.11 WebSphere Commerce Payments configuration 110
3.11.1 Start the WebSphere Commerce Payments Server 110
3.11.2 Add payment information for the store 110
3.12 Deploy a store 110
3.12.1 Download updated model stores from the Web 111
3.12.2 Publish the store from Store Services. 111
3.12.3 Compile the WebSphere Commerce tools JSPs (WCS Tools). . . 113
3.13 WebSphere Commerce runtime verification 113
3.13.1 Verify the WebSphere Commerce administration tools 113
3.13.2 Verify the WebSphere Commerce store 114
3.13.3 Verify WebSphere Commerce Payments functionality 115
3.14 Auto start servers 115

Chapter 4. Implementing a 2-tier remote DB2 server runtime 117
4.1 Planning for the remote DB2 server runtime. 118
4.2 Red Hat Linux installation 119
4.3 DB2 Server installation 119
4.4 DB2 Client installation 120
4.4.1 DB2 pre-installation tasks 120
4.4.2 Install the DB2 Client. 120
4.4.3 Verify the DB2 Client installation. 123
4.4.4 Install DB2 UDB V7 FixPak 7 124
4.4.5 Configure and verify the DB2 Client 125
4.4.6 Configure and verify the DB2 client and server connectivity. 127
4.5 IBM HTTP Server and WebSphere plugin installation 128
4.6 WebSphere Application Server installation 128
4.7 WebSphere Commerce installation. 129
4.8 WebSphere Commerce Payments installation 129
4.9 WebSphere Commerce post-install configuration 130
4.10 WebSphere Commerce instance creation. 130
4.11 WebSphere Commerce post-instance config 131

4.12 WebSphere Commerce Payments configuration 131
4.13 WebSphere Commerce runtime verification 131

Chapter 5. Implementing a 3-tier runtime with the addition of a remote Web server 133
5.1 Planning for the 3-tier remote Web server runtime 134
5.2 Red Hat Linux installation 135
5.3 Install the remote IBM HTTP Server 136
5.3.1 Install the IBM HTTP Server 136
5.3.2 Install the WebSphere plugin for IBM HTTP Server 136
5.3.3 Install WebSphere Application Server V4 FixPak 4 136
5.3.4 Install WebSphere Application Server plugin e-fix 136
5.3.5 Configure the IBM HTTP Server 136
5.3.6 Restart, stop and start the IBM HTTP Server 136
5.3.7 Verify the IBM HTTP Server 136
5.4 Configure the WebSphere plugin 137
5.4.1 Start the WebSphere Application Server 137
5.4.2 Add host aliases to virtual host 137
5.4.3 Remove and update host aliases from virtual hosts 138
5.4.4 Regenerate the Web server plugin 139
5.4.5 Copy plugin-cfg.xml to the remote IBM HTTP Server 140
5.5 Configure Web server for WebSphere Commerce 140
5.5.1 Copy static content to the remote IBM HTTP Server 141
5.5.2 Add entries to httpd.conf for WebSphere Commerce 144
5.5.3 Configure WebSphere Commerce Payments 145
5.5.4 Verify the WebSphere Commerce multi-tier runtime. 146
5.6 WebSphere HTTP transport SSL configuration 147
5.6.1 Create a WebSphere Application Server key database 147
5.6.2 Create a certificate for the WebSphere Application Server. 148
5.6.3 Configure the WebSphere transport 149
5.6.4 Create a IBM HTTP Server plug-in key database. 150
5.6.5 Extract the WebSphere certificate. 150
5.6.6 Import WebSphere certificate as CA on IBM HTTP Server 150
5.6.7 Verify the configuration 151

Appendix A. Linux tips 153
A.1 Red Hat Linux 7.2 installation 154
A.2 Network configuration 157
A.3 Common Linux commands and tasks. 158
A.4 Where to find more information. 160

Appendix B. Automating server startup 161
Auto start overview. 162
DB2 Server auto start. 162

IBM HTTP Server auto start 162
WebSphere Application Server and WebSphere Commerce auto start. 163
WebSphere Commerce Payments auto start. 165

Related publications 167
IBM Redbooks 167
Other resources 167
Referenced Web sites 168
How to get IBM Redbooks 168
IBM Redbooks collections. 168

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:
This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

- IBM®
- Redbooks™
- Redbooks(logo)™
- AIX®
- AS/400®
- CICS®
- Database 2™
- DB2®
- DB2 Universal Database™
- e Strategy™
- Home Director™
- IMS™
- iSeries™
- MQSeries®
- PS/2®
- pSeries™
- RS/6000®
- S/390®
- SecureWay®
- SP™
- Tivoli®
- VisualAge®
- WebSphere®
- xSeries™
- z/OS™
- zSeries™

The following terms are trademarks of International Business Machines Corporation and Lotus Development Corporation in the United States, other countries, or both:

- Lotus®
- Word Pro®
- Lotus QuickPlace ™
- QuickPlace™

The following terms are trademarks of other companies:

ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

C-bus is a trademark of Corollary, Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

SET, SET Secure Electronic Transaction, and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

Other company, product, and service names may be trademarks or service marks of others.

Preface

The Patterns for e-business are a group of reusable assets that can help speed the process of developing Web-based applications and deploying the infrastructure to host the application. This Redpaper will highlight the key elements of the Patterns for e-business and explain how they can be used to provide guidelines for selecting a WebSphere Commerce runtime environment and product mapping.

The focus of this Redpaper is to provide IT specialists with the knowledge to plan and implement a WebSphere Commerce V5.4 for Linux in a single-tier, 2-tier remote Web server or remote database, and 3-tier runtime using DB2 and the IBM HTTP Server.

Included in the implementation procedures are WebSphere Commerce FixPak V5.4.0.3 and WebSphere Application Server V4 FixPak 4. The procedures document best practices, workarounds, and FixPak procedures beyond what is documented in the original release of WebSphere Commerce V5.4.

This Redpaper is a supplement to the *WebSphere Commerce V5.4 Handbook*, SG24-6567 for the Linux platform. The appendixes include tips on Red Hat Linux and auto starting servers.

The team that wrote this Redpaper

This Redpaper was produced by a team of specialists from around the world working at the International Technical Support Organization, Raleigh Center.

John Ganci is a Senior Software Engineer, WebSphere Specialist at the IBM ITSO, Raleigh Center. He writes extensively and teaches classes on WebSphere and related topics. John has 14 years of experience in product and application design, development, system testing, and consulting. His areas of expertise include e-commerce, personalization, pervasive computing, and Java programming. Before joining the ITSO, he developed e-commerce sites for IBM.

Aubrey Shiang is a Consultant I/T Specialist in the WebSphere Commerce arena for IBM ATSS (Americas Technical Sales Support) organization. She has 20 years of experience in software product design, development and integration. She holds a M.S. degree in Computer Science from George Washington University. Her expertise lies in providing multiple facets of support to the WebSphere Commerce brand, including e-commerce architecture analysis and design, configuration support, as well as participating in solutions assurance reviews.

Thanks to the following people for their contributions to this project:

Adrian Warman, IBM UK

Joe DeCarlo, IBM ITSO, San Jose Center

Jakob Carstensen, IBM Linux Solutions Marketing

Mark Ho, IBM Canada

Peter Kovari, IBM ITSO, Raleigh Center

Margaret Ticknor, IBM ITSO, Raleigh Center

Become a published author

Join us for a two- to six-week residency program! Help write an IBM Redbook dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You'll team with IBM technical professionals, Business Partners and/or customers.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you'll develop a network of contacts in IBM development labs, and increase your productivity and marketability.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our papers to be as helpful as possible. Send us your comments about this Redpaper or other Redbooks in one of the following ways:

- Use the online **Contact us** review redbook form found at:

  ibm.com/redbooks

- Send your comments in an Internet note to:

  redbook@us.ibm.com

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. HZ8 Building 662
  P.O. Box 12195
  Research Triangle Park, NC 27709-2195

1

Introduction

The focus of this Redpaper is to provide IT specialists with the knowledge to plan for and implement WebSphere Commerce V5.4 for Linux in a single-tier, 2-tier remote Web server or remote database, and 3-tier runtime using DB2 and the IBM HTTP Server.

Included in the implementation procedures are WebSphere Commerce FixPak V5.4.0.3 and WebSphere Application Server V4 FixPak 4. The procedures document best practices, workarounds, and FixPak procedures beyond what is documented in the original release of WebSphere Commerce V5.4.

This Redpaper is a supplement to the *WebSphere Commerce V5.4 Handbook*, SG24-6567 for the Linux platform. The appendixes include tips on Red Hat Linux and auto starting servers.

1.1 IBM WebSphere Commerce V5.4 for Linux

This section provides a brief overview of the WebSphere Commerce V5.4 product packaging and a listing of where to find more information.

More detailed information on the product offering and features included can be found in the following:

- *Fundamentals Guide, IBM WebSphere Commerce V5.4* product guide PDF
- *WebSphere Commerce V5.4 Handbook*, SG24-6567 redbook

1.1.1 WebSphere Commerce V5.4 product editions

IBM WebSphere Commerce V5.4 runtime product offerings include the following:

- IBM WebSphere Commerce V5.4, Professional Entry Edition

  Targeted at small businesses, it runs on a single-tier Windows 2000 node.

- IBM WebSphere Commerce V5.4, Professional Edition

  This offering is targeted at medium to large companies in need of B2C solutions. This edition is available for Windows NT/2000, AIX, Solaris, Linux on x86 hardware (Red Hat and SuSe), iSeries, z/OS and OS390. It includes support for single-tier or multi-tier enterprise runtime configurations.

- IBM WebSphere Commerce V5.4, Business Edition

  This offering is targeted at medium to large companies in need of B2B and B2C solutions available for Windows NT/2000, AIX, Solaris, Linux on x86 architecture (Red Hat and SuSe), iSeries, z/OS and OS390. This edition is capable of running on a single-tier or on multi-tier enterprise runtime configurations. The Business Edition is a superset, meaning that all features of the Professional Edition are included in Business Edition and it also includes additional functionality targeted towards Business-to-Business (B2B) e-commerce.

1.1.2 WebSphere Commerce Studio V5.4 product offerings

The development tools suite used to create and customize WebSphere Commerce store assets is called WebSphere Commerce Studio. For each runtime product offering (Entry, Professional, Business Edition) there is a corresponding development product offering:

- IBM WebSphere Commerce Studio V5.4, Entry Developer Edition for Windows NT and Windows 2000
- IBM WebSphere Commerce Studio V5.4, Professional Developer Edition for Windows NT and Windows 2000
- IBM WebSphere Commerce Studio V5.4, Business Developer Edition for Windows NT and Windows 2000

The main development tools included are IBM VisualAge for Java V4, Enterprise Edition, WebSphere Studio V4.0, Advanced Edition and WebSphere Commerce Studio extensions.

1.1.3 Finding WebSphere Commerce information

There is a vast amount of information available regarding WebSphere Commerce and WebSphere. Sometimes, the key to solving a problem is knowing where to find the necessary information. This section highlights the key information available in the WebSphere Commerce V5.4 online documentation, the WebSphere Commerce technical library available on the Web, and IBM Redbooks.

WebSphere Commerce V5.4 online documentation

The WebSphere Commerce V5.4 online documentation can be very useful for finding information listed by a specific task, role, or index.

WebSphere Commerce V5.4 technical library

The WebSphere Commerce V5.4 technical library provides a wide array of product user guides, tech notes, readme files and tutorials. The library is available from the following URLs:

- IBM WebSphere Commerce V5.4, Business Edition

  `http://www.ibm.com/software/webservers/commerce/wc_be/lit-tech-general.html`

- IBM WebSphere Commerce V5.4, Professional Edition

  `http://www.ibm.com/software/webservers/commerce/wc_pe/lit-tech-general.html`

The information on the Web site is organized into development phases (getting started, installation, implementation, and operations) and/or by the type of information (user guides, technotes, readme, and tutorials). The information available on this site is updated periodically as needed to reflect updates in new service levels of the product, documentation corrections, and new information.

Note: We recommend that you refer to the URLs given for the WebSphere Commerce technical library to get the most current information.

IBM Redbooks and Redpapers

As many readers already know, Redbooks offer a wealth of information about WebSphere Commerce, WebSphere, DB2, Java programming, and nearly any technical topic or product created by IBM and key business partners.

The IBM Redbook Web site can be found at:

http://www.ibm.com/redbooks

Once on the IBM Redbooks site, you can search for a given topic, select the desired link, and download the IBM Redbook or Redpaper PDF. In addition, some Redbooks and Redpapers provide additional material zip files containing sample code and configuration files.

We have listed some key WebSphere Commerce Redbooks:

- *WebSphere Commerce V5.4 Handbook*, SG24-6567
- *WebSphere Commerce V5.4 Developers Handbook*, SG24-6190
- *e-commerce Patterns for Building B2C Web Sites Using IBM WebSphere Commerce Suite V5.1*, SG24-6180
- *B2B e-Commerce Composite Pattern Using IBM WebSphere Commerce Suite V5.1, Business Edition*, SG24-6181
- *e-commerce Payment Solutions Implementation and Integration Using IBM WebSphere Payment Manager*, SG24-6177
- *Mobile Commerce Solutions Guide using WebSphere Commerce Suite V5.1*, SG24-6171

1.2 Redpaper objectives

The reuse of architectural assets from the Patterns for e-business can save weeks of effort, reduce costs and add to the intellectual capital of the development organization. However, when the solution architect is faced with a complex business problem and disparate customer requirements, applying a pattern may be difficult. Moreover, when the application platform has already been mandated by the customer or the sales process, the application of patterns may be further complicated.

Patterns for e-business are a group of reusable assets that leverage the experience of IBM architects to create solutions for e-business problems. The Patterns for e-business can be broken down into the following groups:

- Business patterns that identify the interaction between users, businesses and data to create simple functionality.
- Integration patterns that connect Business patterns together to create complex functionality.
- Composite patterns that are combinations of Business and Integration patterns for advanced applications.
- Custom designs that are developed to solve specific e-business problems for a single customer organization.

Depending on the customer requirements, the Application and Runtime patterns describe the application and runtime environments needed to develop the solution. The Product mappings and guidelines for the design, development and deployment of the solution are assets for supporting the implementation of the solution. Figure 1-1 illustrates how the assets of the patterns can be used to translate customer requirements into an application and infrastructure for deployment.

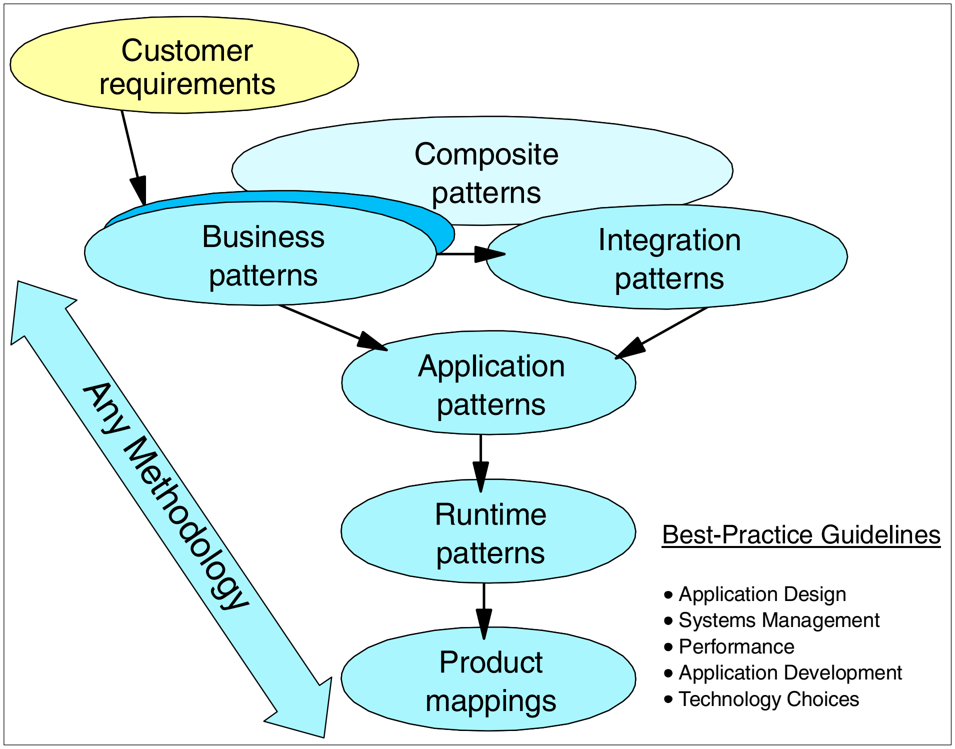


Figure 1-1 Applying the Patterns for e-business

IBM WebSphere Commerce Studio V5.4, Business Edition is a solution for selling in the B2B and B2C environments, with extensions beyond the traditional Web store front. It includes advanced capabilities that enable organizations to build and maintain meaningful relationships with customers and trading partners. The application reduces complexity by managing business processes and integrating them with external and back-end business applications. In addition, Business Edition allows customers and trading partners to collaborate with virtual teaming technology.

This Redpaper will highlight the key elements of the Patterns for e-business and the Sell-Side Hub composite pattern to assist in architecting a B2B solution using Business Edition. We will explain how the patterns can be used to provide guidelines for selecting the WebSphere Commerce runtime environment and product mapping. To demonstrate the reuse of infrastructure and deployment patterns, we have documented detailed procedures to instantiate the runtime product mapping for a WebSphere Commerce 3-tier Linux environment using DB2 UDB V7.2 and the IBM HTTP Server V1.3.19.3.

1.3 Structure of the Redpaper

This Redpaper is structured as follows:

- Chapter 1, “Introduction” on page 1 describes how the Patterns for e-business architectural assets can be reused to speed the process of developing Web-based applications and deploying the infrastructure to host the application. This will ultimately save time and reduce cost.
- Chapter 2, “Working with the e-Marketplace patterns” on page 9 provides a more in-depth understanding of using patterns and features the e-Marketplace patterns to assist in architecting a B2B solution using IBM WebSphere Commerce Studio V5.4, Business Edition.
- Chapter 3, “Implementing a single-tier runtime” on page 31 provides detailed procedures, best practices, workarounds and FixPaks beyond the original release of WebSphere Commerce V5.4, including WebSphere Commerce FixPak V5.4.0.3 and WebSphere Application Server V4 FixPak 4.
- Chapter 4, “Implementing a 2-tier remote DB2 server runtime” on page 117 provides detailed procedures and best practices for implementing a remote DB2 database server for WebSphere Commerce V5.4 runtime environment on Linux.
- Chapter 5, “Implementing a 3-tier runtime with the addition of a remote Web server” on page 133 provides detailed procedures and best practices for adding a remote IBM HTTP Server to an existing single-tier or two-tier remote database server runtime, and for creating a 3-tier runtime environment for WebSphere Commerce V5.4 runtime environment on Linux.
- Appendix A, “Linux tips” on page 153 includes common tasks, tips and procedures used in Red Hat Linux.
- Appendix B, “Automating server startup” on page 161 includes tips on automating the startup of DB2, IBM HTTP Server, WebSphere Application Server, WebSphere Commerce, and WebSphere Commerce Payments.

2

Working with the e-Marketplace patterns

In this chapter, we discuss the major steps in selecting and implementing the e-Marketplace patterns. We start with understanding the idea of e-Marketplaces. Next, we focus on selecting a specific pattern: depending on the pattern, you will need to implement different services. The process of selecting Application and Runtime patterns and mapping the products into the selected runtime environment will conclude the discussion of the e-Marketplace patterns.

For more information the Patterns for e-business, refer to the Patterns for e-business home page at:

    http://www.ibm.com/developerworks/patterns

Also, for an in-depth look at patterns, refer to the book by Jonathan Adams, et al.,*Patterns for e-business: A Strategy for Reuse*, IBM Press, 2001, ISBN 1931182027.

2.1 e-Marketplaces

e-Marketplaces are trading exchanges that facilitate and promote buying, selling, and business communities among trading partners within certain industries. These solutions represent some of the most comprehensive and complex e-business applications that exist today.

There are three types of e-Marketplaces:

- Trading Exchange
- Sell-Side Hub
- Buy-Side Hub

2.1.1 Trading Exchange composite pattern

A Trading Exchange composite pattern allows buyers and sellers to trade goods and services on a public site. The Composite pattern for the Trading Exchange, shown in Figure 2-1, includes the following:

- The Self-Service business pattern that facilitates the interaction between the buyer and the e-Marketplace. Activities such as purchasing from an aggregated catalog, participating in auctions or making exchanges are performed using this pattern.
- The Self-Service business pattern also helps the non-commerce seller perform functions such as updating the catalog, checking orders, checking Request for Quotations and accessing orders.
- The Information Aggregation business pattern is used to create the e-Marketplace catalog from the multiple sources of suppliers' product files, pricing files, advertising literature and so on.
- The Application Integration pattern is used to integrate these two Business patterns seamlessly and also to integrate with existing e-Marketplace support systems, such as billing.
- The Access Integration pattern is used to provide a Portal interface, single sign-on functions, and personalization functions for the e-Marketplace.

In addition to these basic functions, there are many additional functions that can be added to a Trading Exchange as it evolves. For instance:

- The Collaboration business pattern can be used to enable the purchasing approval process.
- The Extended Enterprise business pattern can be used on both the buyer and seller side of a Trading Exchange. On the buyer side, the pattern defines the interaction between the buyer's procurement system and the commerce functions of a Trading Exchange. On the seller side, this pattern defines the

interaction between the procurement functions of a Trading Exchange and its suppliers.

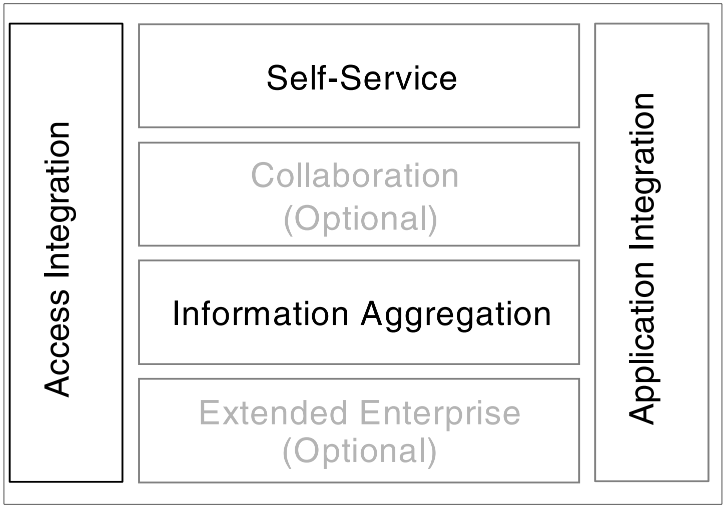


Figure 2-1 Trading Exchange composite pattern

2.1.2 Sell-Side Hub composite pattern

In a Sell-Side Hub composite pattern, the seller owns the e-Marketplace and uses it as a vehicle to sell goods and services to prospective buyers across the Web. The Composite pattern for the Sell-Side Hub, shown in Figure 2-2, includes the following:

- An Access Integration pattern that helps provide a unified customer interface.
- A Self-Service business pattern that allows users to browse through a catalog, create an order, and place an order with the hub.
- An Information Aggregation business pattern that is used to create the Sell-Side Hub catalog from the multiple sources of suppliers' product files, pricing files, advertising literature, and so on.
- Application Integration patterns that integrate the Business patterns that are a part of the Sell-Side Hub composite pattern.

In addition to these basic functions, there can be several variations on this pattern. These include:

- Adding a Collaboration business pattern that allows for and enables auctions, reverse auctions, and other collaborative buying functions.
- Adding an Information Aggregation business pattern that will help integrate and present a unified catalog that combines raw catalog data with expert

advice, product comparisons, and recommendations which can be pulled off public Internet sites.

- Adding an Extended Enterprise business pattern that will integrate the Sell-Side Hub with external service providers, such as a financial institution to handle credit processing, or a shipping company to handle the physical delivery of goods.

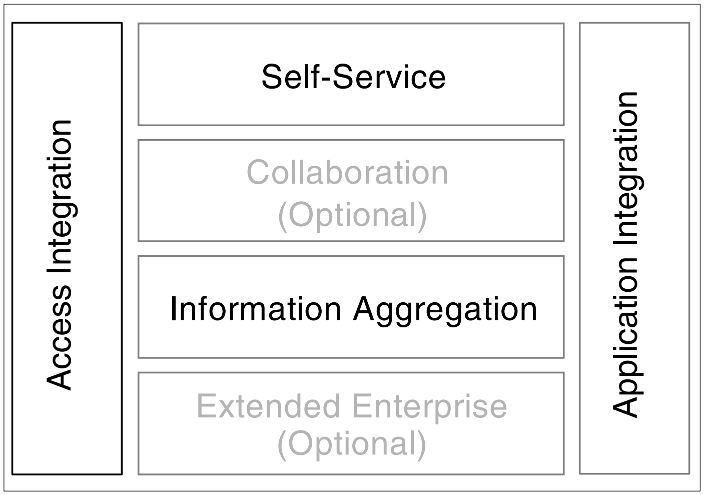


Figure 2-2 Sell-Side Hub composite pattern

2.1.3 Buy-Side Hub composite pattern

In a Buy-Side Hub composite pattern, the buyer of goods owns the e-Marketplace and uses it as a vehicle to leverage the buying or procurement budget to solicit the best deals for goods and services from prospective sellers across the Web. The Composite pattern for the Buy-Side Hub, shown in Figure 2-3, includes the following:

- An Access Integration pattern that helps provide a unified customer sign-on capability and a personalized user interface.
- A Collaboration business pattern that allows users to post bids, participate in auctions and respond to Requests for Proposals (RFP) and Requests For Quotations (RFQ).
- A Self-Service business pattern that allows buyers to create RFQs and RFPs.
- Application Integration patterns that integrate the Buy-Side Hub with procurement systems and other core business applications.

In addition to these basic functions, there can be several variations on this pattern. These include:

- Adding an Information Aggregation business pattern which will help integrate content sources across the Web.
- Adding an Extended Enterprise business pattern which will integrate the Buy-Side Hub with external service providers, such as financial institutions.

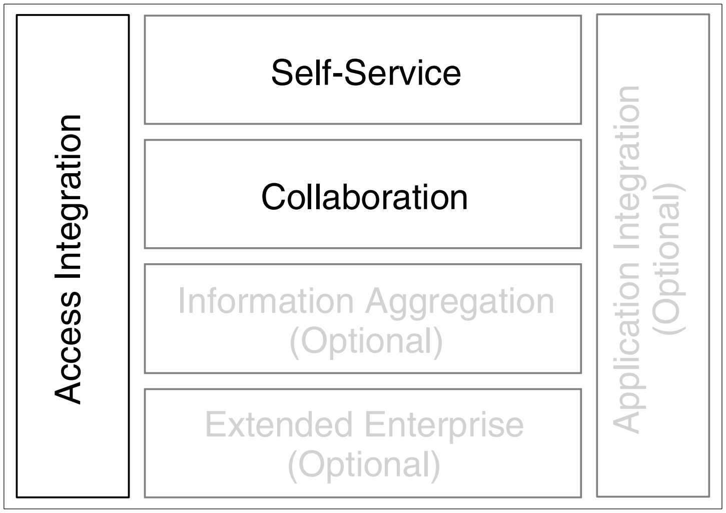

Access Integration
Self-Service
Collaboration
Information Aggregation (Optional)
Extended Enterprise (Optional)
Application Integration (Optional)

Figure 2-3 Buy -Side Hub composite pattern

2.2 Selecting the Application pattern

An e-Marketplace serves as a hub that brings together buyers and sellers. Marketplaces can provide a unified view of the goods and services traded in the market and a variety of mechanisms to help buyers and sellers trade these products.

Implementing a successful e-Marketplace requires that you identify an appropriate Application pattern to meet the requirements of buyers and sellers using the solution.

An Application pattern is a high-level view of the principal layout of the application. It does not show middleware, files, or databases, nor does it describe the detailed application design.

Just as an e-Marketplace is built up from e-commerce, user service, and application integration building blocks, the e-Marketplace patterns are built from a combination of Business patterns and Integration patterns corresponding to these building blocks.

The Business patterns and Integration patterns that serve as component patterns in the e-Marketplace patterns are as follows:

- Self-Service
- Information Aggregation
- Collaboration
- Extended Enterprise
- Access Integration
- Application Integration

Depending on the type of e-Marketplace being deployed (a Trading Exchange, Buy-Side Hub or Sell-Side Hub), different patterns are combined to implement these various functionalities.

2.2.1 Application pattern 1: e-commerce

The Electronic Commerce composite pattern, as applied to the e-Marketplace composite pattern, embodies the commerce interaction undertaken by a purchaser in the e-Marketplace. An e-Marketplace contains interactions, such as RPQ/RFP and exchange trading, that go beyond what is typically encountered in a B2C online buying scenario. Figure 2-4 shows the matching and selling functions of the marketplace.

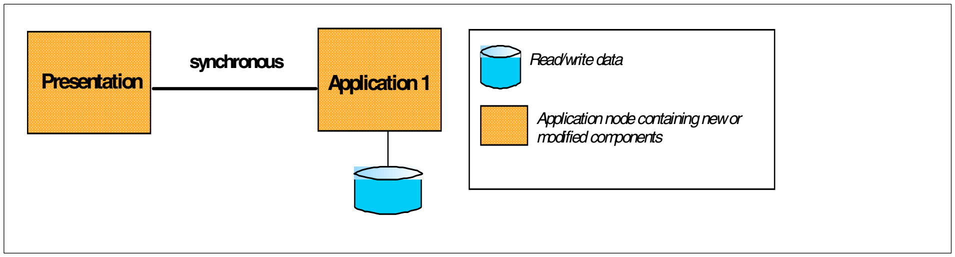


Figure 2-4 e-commerce Application pattern

2.2.2 Application pattern 2: Self-Service

The Self-Service business pattern addresses non-purchasing interactions with the marketplace, such as a supplier checking order statistics and providing catalog updates. In Figure 2-5, Application 2 is a content creation application or an application to provide supporting services such as accessing purchase orders.

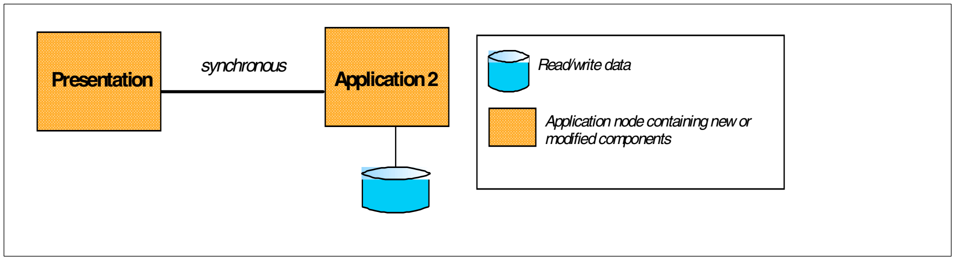


Figure 2-5 Self-Service Application pattern

2.2.3 Application pattern 3: Collaboration

The Collaboration business pattern comes into play in approval workflow, where an approver must sign off on a purchase before it is submitted to the supplier. In Figure 2-6, Application 3 is a workflow application for implementing such flows.

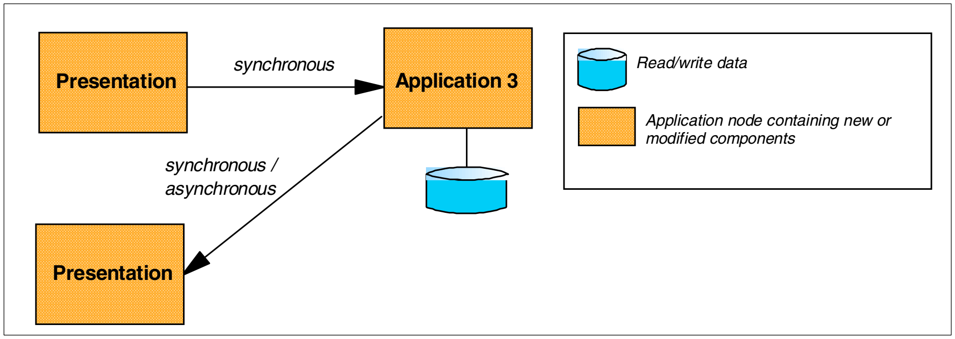

Presentation
synchronous
Application 3
Read/write data
Application node containing new or modified components
synchronous / asynchronous
Presentation

Figure 2-6 Collaboration Application pattern

2.2.4 Application pattern 4: Batch data exchange

This component of an e-Marketplace solution design does not correspond to one of the high-level Business patterns. Rather, it functions as a batch data exchange, such as for the programmatic import of catalog information into the aggregate catalog of the e-Marketplace. This is a bulk transfer of information, as opposed to an automated data exchange following a managed Business-to-Business protocol. Application 4 in Figure 2-7 provides aggregation and publishing of the catalog in the marketplace. It is not necessarily the same as the application that manages the catalog.

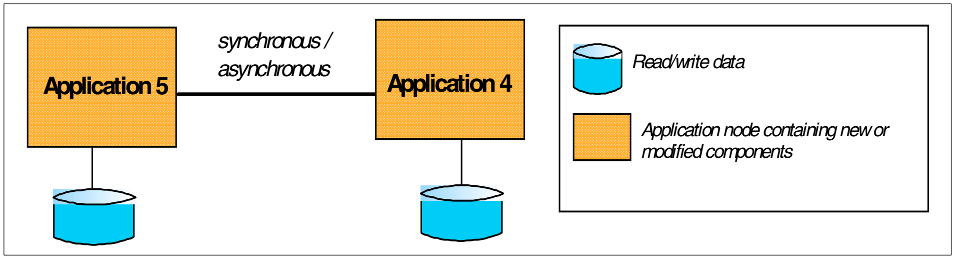
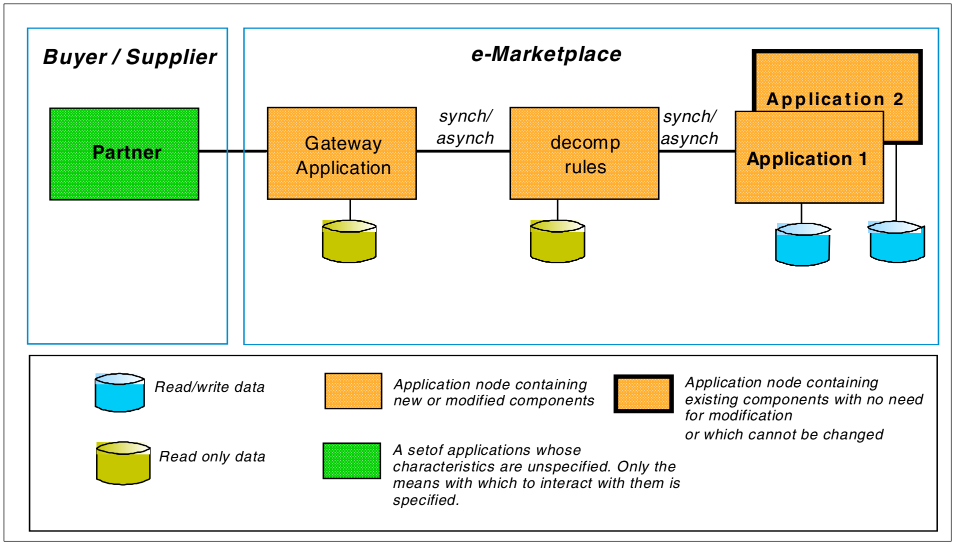

Application 5
synchronous / asynchronous
Application 4
Read/write data
Application node containing new or modified components

Figure 2-7 Batch data exchange Application pattern

2.2.5 Application pattern 5: Extended Enterprise

Component application pattern 5 defines the interaction between the e-Marketplace and the supplier's commerce system, as well as that between the buyer's procurement system and the commerce functions of the e-Marketplace, where this interaction is governed by a well-defined and executable contract. The Application 1 node in this design is the same Application 1 in component Application pattern 1. When the commerce functions of the e-Marketplace integrate with automated buyer or supplier systems, this pattern is applied. Other applications in the marketplace might also need to interact with partner applications.



Buyer / Supplier
e-Marketplace
Partner
Gateway Application
synch/ asynch
decomp rules
synch/ asynch
Application 2
Application 1
Read/write data
Read only data
Application node containing new or modified components
A setof applications whose characteristics are unspecified. Only the means with which to interact with them is specified.
Application node containing existing components with no need for modification or which cannot be changed

Figure 2-8 Extended Enterprise Application pattern

Subsets of the full e-Marketplace function

An e-Marketplace involves many types of interactions with buyers and sellers. Some of these can be user-driven and others can be carried out programmatically. The degree of programmatic integration of buyers and sellers into the e-Marketplace is a feature that can be used to classify subsets of the full solution design defined above. These subsets can represent a phased approach to implementing the e-Marketplace or a means of addressing specific requirements for B2B commerce. For more information on the subsets, refer to *B2B e-commerce Using WebSphere Commerce Business Edition, Patterns for e-business Series*, SG24-6181.

2.3 Selecting the Runtime patterns

The available Runtime patterns depend largely on the integration requirements, arising from the potential interactions between multiple buyers and suppliers. Development of an e-commerce site is normally iterative with the addition of extra participants and degrees of automation as the site evolves. This evolution should be planned for from the beginning and the site may implement different Application patterns progressing from simpler to more complex patterns with the addition of extra nodes and interactions to implement this.

This section focuses on the logical nodes and their interaction in each of the Runtime patterns. The network pattern may vary, depending on enterprise security policy regarding the processing and storage of sensitive data. Other Non Functional Requirements such as availability/scalability may require one logical node to be represented by more physical nodes to provide redundancy and improve response times under high load. Conversely, in development or system test environments, separate logical nodes may be consolidated into one physical node.

The deconstruction of the problem into separate nodes provides an initial picture that can be used for a walkthrough to verify the design, initial cost and planning of the selected solution.

Most references to vertical scalability in this chapter assume upgrading the power, number or memory capacity of the processors for a given platform. Please bear in mind that one of the benefits of the IBM Software Strategy for e-business is that vertical scalability is also possible by migrating the application to other supported platforms, such as IBM eServer zSeries (S/390), pSeries (RS/6000), and iSeries (AS/400).

A more complete discussion of sizing and scalability as applied to WebSphere Commerce may be found in the following documents:

- The *IBM WebSphere Commerce Suite Capacity Planning Guide V5.1* at:

  `http://ibm.com/software/webservers/commerce/wcs_pro/WCSCapacityPlanningGuide.pdf`

- The redbook *WCS V5.1 Performance Tuning*, SG24-6258, found at:

  `http://www.redbooks.ibm.com/pubs/pdfs/redbooks/sg246258.pdf`

2.3.1 Runtime patterns

In this section, we present the basic high-level Runtime patterns that are predominant in many existing e-commerce implementations. These patterns are becoming known as *proven Runtime patterns* because they have existed for some time and are implemented in many production sites. These proven patterns represent the core Runtime patterns generally considered at the core of an e-Marketplace. It is important to note that the additional subsets presented in this chapter all inherit from this foundation and extend it appropriately.

Basic Runtime pattern

The basic Runtime pattern consists of all of the software components of WebSphere Commerce Business Edition V5.4 for Solaris installed on a single system, as shown in Figure 2-9.

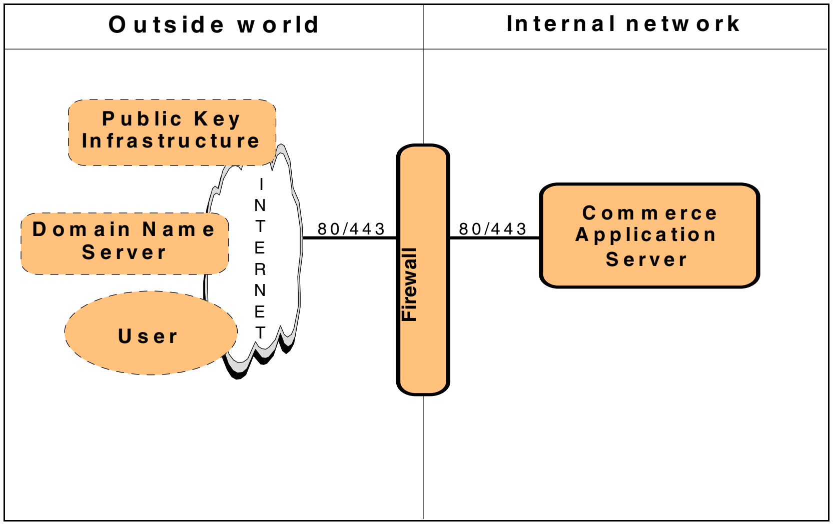


Figure 2-9 Basic Runtime pattern

While this Runtime pattern is simple to implement, it has some limitations such as the lack of separation between the Web server and the application server. A common solution to this problem is to place the application server behind the domain firewall, where it benefits from the extra security provided by the firewall. You should also be aware that there are limitations to the performance gains achieved through the addition of Web and application servers and that it is largely dependent on the extent of the workloads of back-end resources and other subsystems.

Basic Runtime pattern - variation 1a

This Basic Runtime pattern, variation 1a, is commonly implemented as the foundation for e-commerce sites and is usually extended in line with the load and functionality requirements of the site. Figure 2-10 shows a diagram of this Runtime pattern.

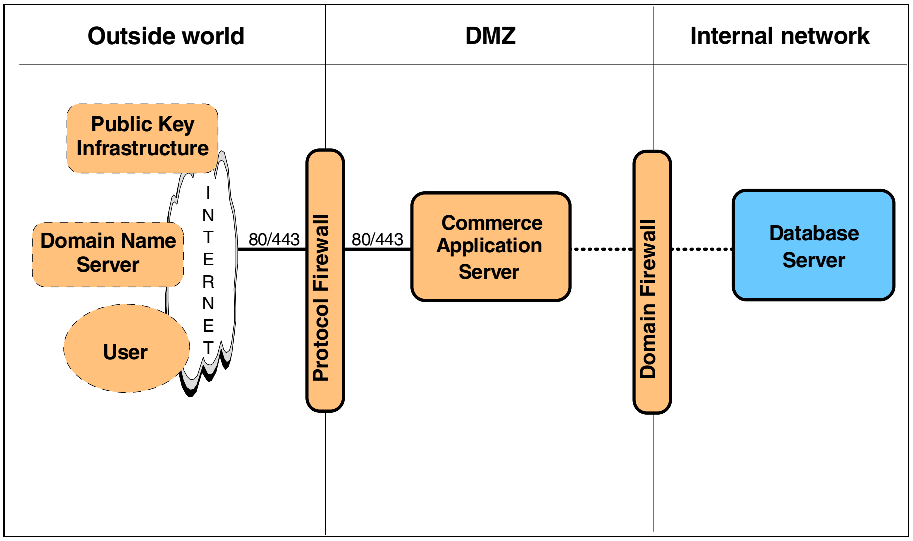


Figure 2-10 Basic Runtime pattern - variation 1a

In this pattern, the Web server and the application server are combined to form the commerce server node residing within the Demilitarized Zone (DMZ). This is separated by a protocol firewall from the outside world and a domain firewall to protect the directory services and primary data repository. Most of the application logic is executed on the commerce server node under the application server and served to the client through the Web server.

While this Runtime pattern is simple to implement, it has a number of inherent limitations, such as:

- Limited availability and fail-over capability.
- No support for horizontal scalability, as there is only one application server.
- Limited vertical scalability options. However, additional processing power and memory can assist in this area.
- Security for the application server applications is limited to that provided by the protocol firewall. While this is often acceptable for static HTML content residing on the Web server, it is generally inappropriate for securing critical applications.
- The number of simultaneous connections to the Web server is restricted to its capacity.

This Runtime pattern is intended for database intensive applications, not only increasing the application data security, but also improving the performance by concentrating the database processing exclusively on a separate machine.

There are many alternatives to increasing the security in the DMZ, but they are beyond the scope of this Redpaper.

Basic Runtime pattern - variation 1b

The pattern and variations described above may prove to be a constraint for e-Marketplaces that are very large or very sophisticated. For those e-Marketplaces, some of the nodes in the pattern must be split into multiple nodes with expanded roles and new nodes will need to be introduced. Figure 2-11 shows the Basic Runtime pattern - variation 1b.

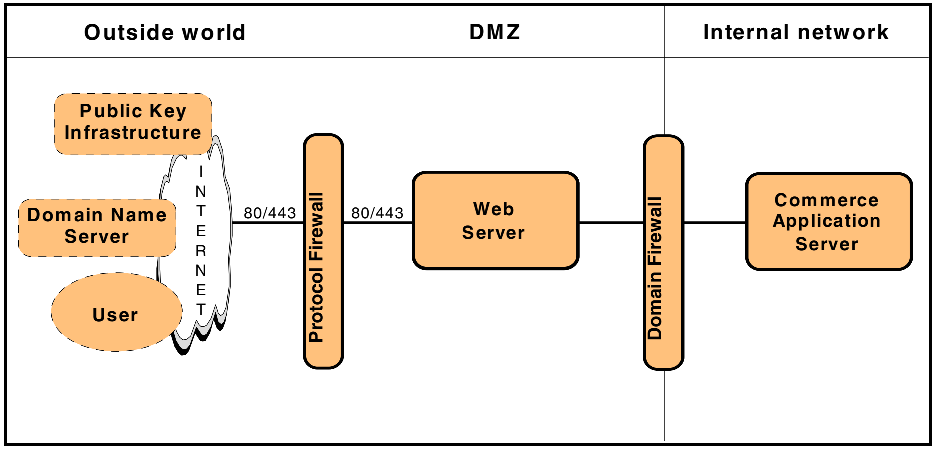


Figure 2-11 Basic Runtime pattern - variation 1b

When choosing this Runtime pattern, you can increase the security in the scenario by implementing firewalls to protect the application server and data and creating a Demilitarized Zone (DMZ) in which to place the Web server.

This remote Web server configuration will allow all the application's static content (that is, HTML and graphics files) to be served on the Web server machine, separate from the application server, commerce server and database server; from a performance perspective, this scenario allows for the creation of a Web server cluster in the DMZ to handle major amounts of concurrent hits on your site, concentrating the Java processing exclusively on a separate machine.

The potential downside to this configuration is that if the application is database intensive, CPU performance may not be as high as if the database and application server were running on separate servers.

There are alternatives to increase the security in the DMZ, performance and work load management, but they are beyond the scope of this Redpaper.

Basic Runtime pattern - variation 2

This Runtime pattern is set up for performance, scalability, and security reasons. Having the three separate servers, as seen in Figure 2-12, allows for the placement of firewalls between critical servers and also allows for the addition of servers to either of the three tiers with the least amount of trouble.

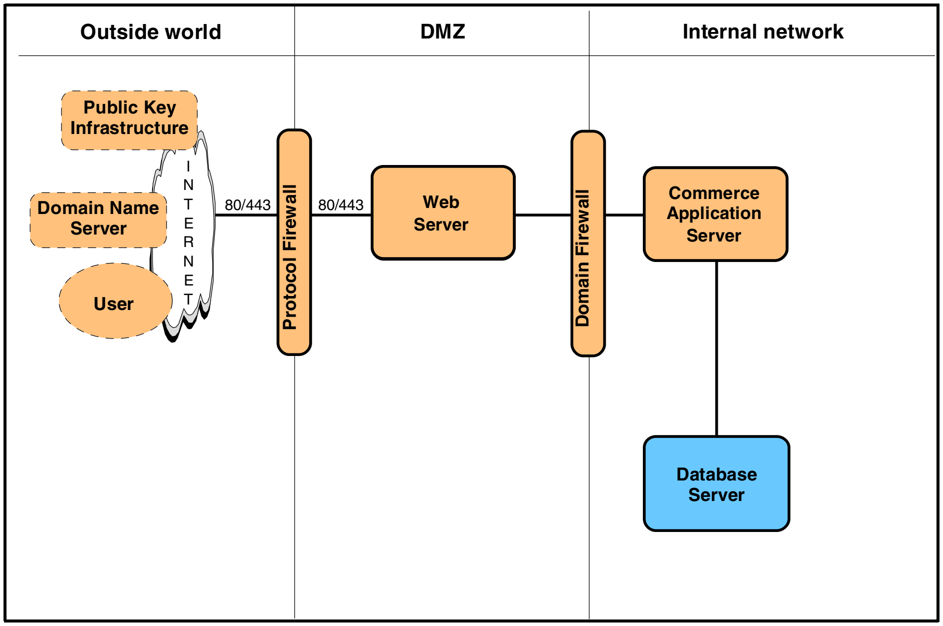


Figure 2-12 Basic Runtime pattern - variation 2

This configuration separates the Web server and places it into a DMZ for serving static content; this optimizes the static content serving when your site has large amounts of concurrent hits; it also facilitates scalability for creating a Web server cluster to handle those requests. A load balancer such as IBM Network Dispatcher is required to manage that Web server cluster.

Applying the same concepts to the Commerce Application Server and the Database Server, you can concentrate the Java processing on one machine and the Database processing on another, separate machine.

To summarize, the Runtime pattern shown in Figure 2-12 offers high Java processing performance in database intensive applications for large amounts of concurrent transactions.

Figure 2-13 represents a variation to any of the above discussed Runtime patterns and can be applied to offer high availability.

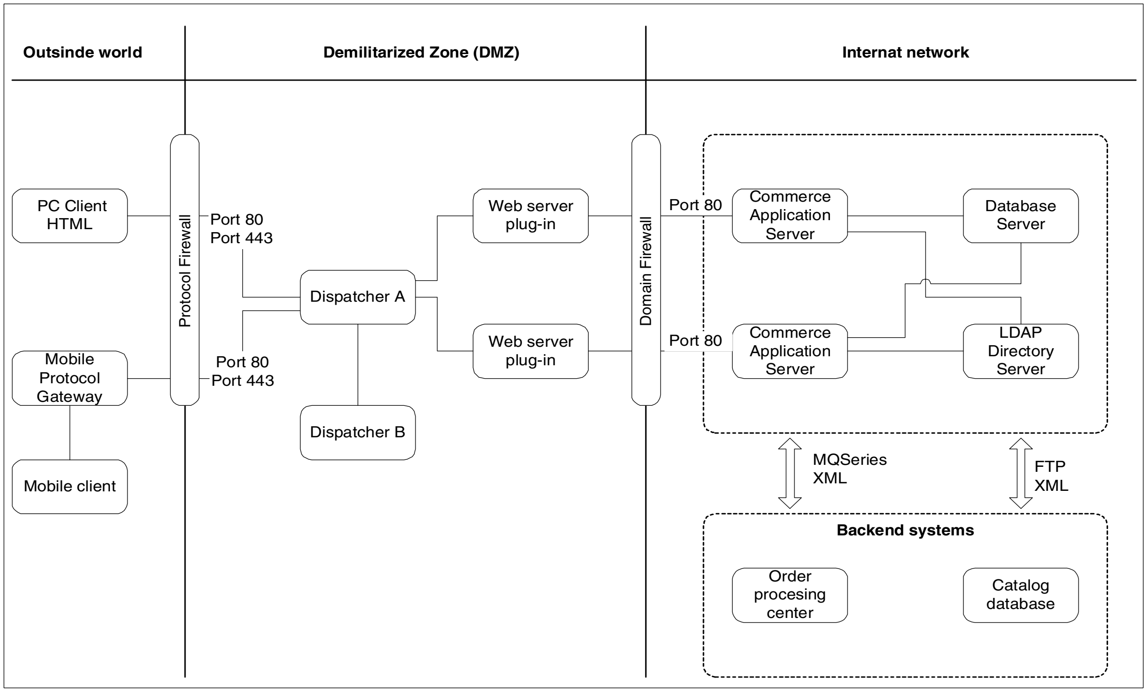


Figure 2-13 Runtime pattern tiers and nodes

The Runtime patterns are shown in graphical form. Each pattern will consist of several nodes, describing the function represented on that node. Most patterns will consist of a core set of common nodes, with the addition of one or more nodes unique to that pattern.

Let us take a closer look at the Runtime node definitions.

2.3.2 Runtime node definitions

This section describes the nodes shown in Figure 2-13.

Outside world

As shown in Figure 2-13, the first tier is the outside world (Internet). The Internet zone contains nodes such as the public key infrastructure, domain name servers and, of course, clients.

Client nodes

Client nodes are responsible for accepting and validating the user input, communicating the user input to the commerce server node, and presenting the results received from the commerce server node to the user. Clients use HTTP to communicate with the Web application server. Web clients include the following types.

- Browser-based clients

  A Web browser such as Microsoft Internet Explorer or Netscape Navigator uses the HTTP and HTTPS protocols to communicate with the Web server. These clients display HTML and DHTML Web pages. In addition, they are capable of processing client-side JavaScript for enhancing navigation and performing simple input validation and simple error handling. Note that the use of JavaScript validation to reduce network traffic still requires that the same validation be performed server-side for security reasons. Session management normally uses cookies if these are supported and enabled on the browser, although WebSphere Commerce can be configured to use URL rewriting. The browser should support SSL to allow the encryption of sensitive data that is transferred over a public network. Furthermore, the majority of browsers can display small Java applets to enhance the GUI.

- Mobile clients

  Mobile clients or wireless clients include mobile phones, wireless PDAs, and wireless laptops. The two dominating wireless protocols are WAP and i-mode, which use Wireless Markup Language (WML) and compact HTML (cHTML) respectively to display content on the device. Wireless clients connect to the wireless network via a wireless network provider protocol gateway. The request from the mobile device is first processed by the wireless protocol gateway, which then converts the request to HTTP or HTTPS protocol to access the Web server.

- Application clients

  Application clients are primarily Java applets or Java applications. These clients provide rich graphical user interfaces compared to HTML clients. They may communicate with the Web application server over a number of protocols, including HTTP and IIOP.

The advantages of using Web clients are as follows:

- The majority of the presentation logic and all of the business logic will reside on the server.
- The client part of the application is lightweight and downloads quickly.
- It is easier to secure, scale and maintain presentation and business logic that resides on the server.

External systems

These include systems internal to the purchasing or supplier organizations that communicate automatically with the e-Marketplace, for example when a supplier delivers catalog information to the content management node. These connections may use a VPN, a WAN or TCP/IP with some form of encryption.

Public Key Infrastructure (PKI) node

A Public Key Infrastructure (PKI) node is a collection of standards-based technologies and commercial services to support the secure interaction of two unrelated entities (for example, a public user and a corporation) over the Internet. In the context of the patterns defined in this Redpaper, PKI supports the authentication of the server to the browser client, using the SSL protocol.

Mobile gateway

The Mobile gateway provides a connection to the wireless network. It may perform or forward an authentication request depending on the device ID. It may also assist session management by allocating an IP number to the device and handling cookies on behalf of the device.

Mail server

e-mail notifications are often sent at several points in order processing. The mail server on this node uses SMTP. The address of this machine can be configured in instance creation.

Demilitarized Zone (DMZ)

The first tier and the second tier are separated by the protocol firewall. The second tier, known as a Demilitarized Zone (DMZ), is a network area that is exposed to an non-trusted outside world (the Internet), and is considered a high-risk zone for attacks. No confidential material should reside on any server in a DMZ in an unprotected form. Only approved ports can be opened between the first and second tier.

Web server node

The Web server node is typically designed for access by HTTP clients and often works in combination with an application server. The Web server serves static content such as HTML and images to Web browser clients. Examples of Web servers include the IBM HTTP Server and Netscape iPlanet Web Server. This node would be provided by the company, on company premises, or hosted inside the enterprise network and inside a Demilitarized Zone (DMZ) for security reasons. In most cases, access to this server would be in secure mode, using services such as SSL, Secure VPN, or IPSec.

Web server redirector node

In order to separate the Web server from the application server, a so-called Web server *redirector* node (or just redirector for short) is introduced. The Web server redirector is used in conjunction with a Web server. The Web server serves HTTP pages and the redirector forwards servlet and JSP requests to the application servers. The advantage of using a redirector is that you can move the application server behind the domain firewall into the secure network, where it is more protected than within the DMZ.

The redirector can be implemented, for example, by either a reverse proxy server or a Web server plug-in such as the servlet redirector function of IBM WebSphere Application Server, Advanced Edition.

Edge Server node

The Dispatcher node provides the following fundamental functions for Web applications:

- It ensures that the application service remains available. This is achieved by providing multiple instances of the service, running on several servers. In the event that a server (or application) becomes unavailable (planned or as the result of a failure), the dispatcher will distribute work only to the remaining operational server. When the server becomes available again, it is added to the pool.
- It ensures that a consistent response time is enjoyed by the client. Statistics on the response times from all servers in a pool can be used to measure the level of acceptable service for the customer. The dispatcher will distribute work evenly across all servers. This will ensure that no one server is overloaded. It can also ensure that no single network path is overloaded when there are other routes available (where servers are split across multiple sub-networks).
- Additionally, the dispatcher can be used as a systems management tool to dynamically add (or remove) instances of an application service from the pool in response to capacity/performance forecasting or for systems maintenance.

  Since the dispatcher is a single point of failure, it should be backed up via heartbeat or HACMP. The heartbeat alternative is usually less expensive and easier to install than HACMP and is available on several platforms.

B2B gateway

This node handles the messaging between the External systems node and the application system, in case the communication protocol is not HTTP.

Internal network

The internal network (intranet), or third tier, is provided by an additional firewall called a domain firewall. No direct traffic from the first tier is allowed to the third tier. The intranet is considered a safe zone for applications that obtain confidential material.

Application server node

This node provides the infrastructure for application logic and may be part of a Web application server node. It is capable of running both presentation and business logic but generally does not serve HTTP requests. When used with a Web server redirector, the application server node will run both presentation and business logic. In other situations, it may be used for business logic only.

The responsibilities of the application server include: receiving requests from the clients, selecting and executing the appropriate business logic based on these requests, coordinating with external services (for example, the LDAP directory) to retrieve data and execute external applications, and finally formulating the response and dispatching it back to the client. To meet these requirements, the application servers provide a range of dynamic page construction, business logic processing, data access, external application integration, session management, load balancing, and fail-over services.

- WebSphere Application Server supports Java-based technologies such as Java Servlets and JavaServer Pages (JSPs) for dynamic page construction.
- Business logic services provide a robust environment for processing business logic independent of the user interface client types. WebSphere Application Server supports components based on technologies such as JavaBeans and Enterprise JavaBeans (EJBs) for programming business logic.
- Connectors are components that support the communication between the application server and the external services such as databases, legacy applications and business partner applications. WebSphere Application Server, Advanced Edition provides a number of connectors including JDBC drivers, JNDI class libraries (used by for LDAP Directory), CICS connectors, MQSeries connectors, and IMS connectors.
- Session management services are necessary because inherently, HTTP is a stateless protocol. The two main types of session management are cookie-based and URL rewriting.
- Security services include authentication, authorization, data integrity, privacy (encryption) and non-repudiation services. The application server provides these services by supporting industry standard protocols such as SSL and LDAP.

- System management services provide a robust runtime environment for the application hosted in the Web application server. These services allow load balancing, fail-over support, remote monitoring, and so on.
- External services include enterprise data sources, existing or new enterprise applications (for example ERPs, financial systems, etc.), and business partner systems.

Commerce application server node

This node provides the infrastructure for the presentation and business logic and may be part of a commerce application server node.

Application database server node

WebSphere Commerce is implemented using a relational database. The database server runs at least the WebSphere Application Server database and the WebSphere Commerce production database. The WebSphere Commerce database contains all information that is used by the WebSphere Commerce application, including information about individual shoppers, items, prices, and so on. The WebSphere Application Server database contains all configuration information for the application server.

The WebSphere Commerce application may use either DB2 Universal Database or Oracle. The DB2 Universal Database ships with the WebSphere Commerce product package.

The database server node's function is to provide a persistent data storage and retrieval service in support of the user-to-business transactional interaction. The data stored is relevant to the specific business interaction.

It is important to note that the mode of database access is perhaps the most important factor determining the performance of this Web application, in all but the simplest cases. Since the WebSphere Application Server and WebSphere Commerce databases have different data profiles, it may be desirable to separate these databases for performance tuning.

Directory and security node

The directory node is a centralized, enterprise-wide directory structure optimized for the rapid retrieval of search results. It normally provides a separate multi-tier user administration function for updating the directory.

WebSphere Commerce supports three methods for authentication and storage of profile information. It can store the information internally, provide integration with SecureWay Directory using LDAP or, alternatively, can be configured to use an alternative service. The connections to this alternative service will need to be custom coded.

The Directory and security services node supplies information about the location, capabilities and various attributes (including user ID/password pairs and certificates) of resources and users known to the Web application system. The node may supply information for various security services (authentication and authorization) and may also perform the actual security processing, for example, to verify certificates. The authentication in most current designs validates the access to the WebSphere Application Server and the database server.

Protocol firewall and domain firewall nodes

Firewalls provide services that can be used to control access from a less trusted network to a more trusted network, with the following traditional implementations:

- Screening routers (the protocol firewall in this design)
- Application gateways (the domain firewall)

The two firewall nodes provide increasing levels of protection at the expense of increasing computing resource requirements. The protocol firewall is typically implemented as an IP router, while the domain firewall is a dedicated node.

2.4 Selecting the Product mappings

The first step in performing a Product mapping is to choose the right platform. The platform chosen should fit into the customer's environment and provide quality of service, such as scalability and reliability, so that the solution can grow with the e-business. When selecting a platform, some of the factors to consider include:

- Existing infrastructure, systems and platform investments
- Available customer and developer skills
- Customer preference
- Quality of service, such as scalability and reliability

Figure 2-14 provides an example Product mapping for the basic Runtime pattern variation 2 using Linux. Figure 2-14 illustrates the software product names and versions implemented.

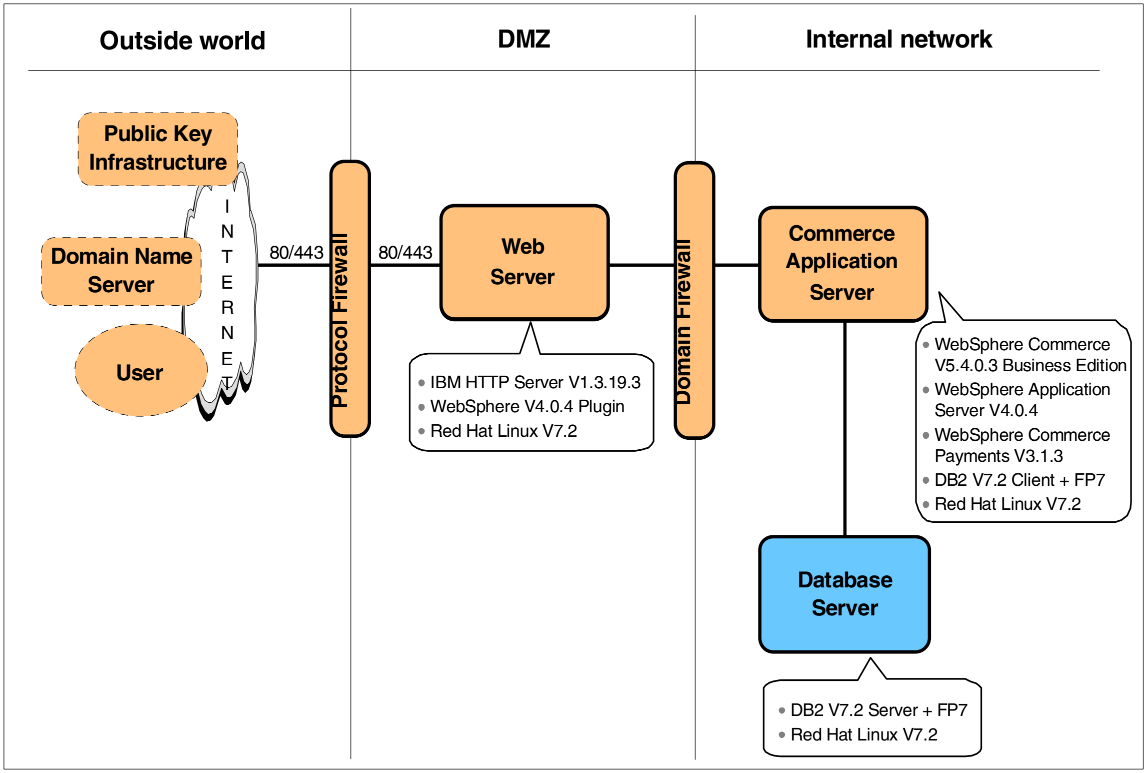


Figure 2-14 Basic Runtime pattern - variation 2 Product mapping - Linux

For a detailed description of the products and versions used in Figure 2-14, refer to 3.1.3, “Software used in the ITSO test environment” on page 34.

2.5 Summary

The IBM Patterns for e-business are a collective set of proven architectures. This repository of assets can be used by companies to facilitate the development of Web-based applications. They help an organization understand and analyze complex business problems and break them down into smaller, more manageable functions that can then be implemented.

3

Implementing a single-tier runtime

This chapter provides detailed instructions for implementing IBM WebSphere Commerce V5.4, Business Edition for Linux in a single-tier runtime environment. The procedures documented include best practices, tips, workarounds and FixPaks beyond the levels that are documented in the product installation guides.

The single-tier scenario documented in this chapter is used most often by developers and IT specialists during the development and test phases. This type of configuration should only be used sparingly in a production environment as it does not offer some of the advanced security and scalability options provided by two- or three-tier configurations.

> **Note:** The scenario complements the procedures documented in the *Installation Guide, IBM WebSphere Commerce V5.4 Professional and Business Edition for Linux* product guide by providing real world examples, best practices, tips, and workarounds.

This chapter includes the following topics:

- Planning
- Red Hat Linux installation
- DB2 Server installation
- IBM HTTP Server and WebSphere plugin installation
- WebSphere Application Server installation
- WebSphere Commerce installation
- WebSphere Commerce Payments installation
- WebSphere Commerce post-install configuration
- WebSphere Commerce instance creation
- WebSphere Commerce post-instance config
- WebSphere Commerce Payments configuration
- Deploy a store
- WebSphere Commerce runtime verification
- Auto start servers

3.1 Planning

This section defines the hardware and software used when implementing the WebSphere Commerce V5.4 for Linux in the ITSO test environment and can be used as a reference for planning.

This section includes the following topics:

- Hardware and software prerequisites
- Hardware used in the ITSO test environment
- Software used in the ITSO test environment

3.1.1 Hardware and software prerequisites

For detailed information on the hardware requirements of IBM WebSphere Commerce V5.4, Business Edition, refer to the following documentation:

- *Installation Guide, IBM WebSphere Commerce V5.4 Professional and Business Edition for Linux*, found at:

  `http://www.ibm.com/software/webservers/commerce/wc_be/lit-tech-general.html`

- IBM DB2 Universal Database

  `http://www.ibm.com/cgi-bin/db2www/data/db2/udb/winox2unix/support/v7pubs.d2w/en_main`

- IBM WebSphere Application Server

  `http://www.ibm.com/software/webservers/appserv/doc/latest/prereq.html`

3.1.2 Hardware used in the ITSO test environment

We used the following hardware within the ITSO test environment:

- IBM eServer xSeries 230 (8658-61Y)

  This system type was used for the single-tier node, and the commerce application server node when used with a remote DB2 server for the multi-tier configurations.

  - 1 GHZ Pentium III CPU
  - 1 GB RAM
  - 18 GB hard disk
  - 1 IBM Ethernet Adapter

- IBM NetFinity 3000 (8476-41U)

  This system type was used for the remote IBM HTTP Server and WebSphere plugin node for the multi-tier configurations.

  - 450 MHZ Pentium II CPU
  - 384 MB RAM
  - 4 GB hard disk
  - 1 IBM Ethernet Adapter

3.1.3 Software used in the ITSO test environment

We used the following software in the ITSO test environment:

- Red Hat Linux 7.2 Standard Edition plus prerequisites
- IBM DB2 Universal Database V7.2.1, Enterprise Edition for Linux plus IBM DB2 UDB V7 FixPak 7 (Web download)
- IBM HTTP Server V1.3.19.3 for Linux (Web download)
- IBM WebSphere Application Server V4.0.1, Advanced Edition for Linux plus IBM WebSphere Application Server V4 FixPak 3 (V4.0.3, Web download)
- IBM WebSphere Commerce V5.4, Business Edition for Linux plus IBM WebSphere Commerce FixPak V5.4.0.3 (Web download)
- IBM WebSphere Payment Manager V3.1.2 for Linux plus IBM WebSphere Commerce Payments FixPak V3.1.3 (Web download)

Note: WebSphere Payment Manager V3.1.2 has been replaced by IBM WebSphere Commerce Payments V3.1.3, which also includes fixes.

Product installation directories

Table 3-1 describes variable names, default installation directories, and components. We will reference these variables throughout this document.

Table 3-1 Product installation directories

Variable	Default directory	Component
<db2_home>	/usr/IBMdb2	DB2 Server
<http_home>	/opt/IBMHTTPServer	IBM HTTP Server
<was_home>	/opt/WebSphere/AppServer	WebSphere Application Server
<wc_home>	/opt/WebSphere/CommerceServer	WebSphere Commerce
<wpm_home>	/opt/WebSphere/PaymentManager	WebSphere Payment Manager

Installation variables

Table 3-2 describes the variable names and values used during the installation.

Table 3-2 Common installation settings

Variable	Default value	Description
<db2_instance_owner>	db2inst1	DB2 Server
<db2_instance_path>	/home/db2inst1	DB2 Server

Users and groups

In our setup, we added the users and groups listed in Table 3-3 during the process of installing the software components.

Table 3-3 Users and groups

User	Group	Home	Shell
root	root	/root	/bin/bash
admin[a]	admin	/home/admin	/bin/bash
db2fenc1	db2fadm1	/home/db2fenc1	/bin/bash
db2inst1	db2iadm1	/home/db2inst1	/bin/bash
db2as	db2asgrp	/home/db2as	/bin/bash
wasuser	wasgroup	/home/wasuser	/bin/bash

[a] Red Hat Linux 7.2 will refuse remote logons by the root user, so you must have a user other than root to log in with. The good news is that you can do this during the installation of Red Hat.

The last three users, db2fenc1, db2inst1, and db2as, are created as part of the DB2 installation. If you should choose to install with one of the other supported databases, you will need to follow its guidelines for database user accounts.

3.2 Red Hat Linux installation

This section describes Linux installation settings for the ITSO test environment.

Note: Appendix A, “Linux tips” on page 153 provides the detailed procedures we used to install Red Hat Linux 7.2 for the ITSO test environment.

3.2.1 Where to find more information on Linux

There are many sources of information on how to install Linux. This section will outline the steps taken to build our single-tier environment. For more detailed instructions on possible installation planning, please refer to these other sources:

- IBM ITSO Redbooks site:

  `http://www.redbooks.com/`

Tip: Search on Linux.

- Red Hat home page:

  `http://www.redhat.com/`

- Red Hat Linux 7.2 gotchas and workarounds:

  `http://www.redhat.com/support/resources/gotchas/7.2/`

- *The Official Red Hat Linux x86 Installation Guide* (V7.1), found at:

  `http://www.redhat.com/docs/manuals/linux/RHL-7.2-Manual/install-guide/`

3.2.2 Linux kernel

The WebSphere Application Server V4.0.1, Advanced Edition for Linux requires a minimum release of 7.2 when using Red Hat Linux.

3.2.3 File systems

During the interactive phase of the Red Hat Linux 7.2 operating system installation, you will need to customize the file systems and allocate the necessary space for the software components installed. In newer versions of Red Hat Linux, the file systems are referred to as *partitions* (as in Windows).

When configuring partitions, there are a couple of options to consider:

- One large root file system - partition (development or test system)

  When configuring a development or test system, you may not be concerned about having unique files systems. In this case, create a swap file approximately one or two times the amount of RAM, and the remainder of the disk will be allocated to / (root) using the EXT file system type.

- Individual file systems (production)

  On a production system, it may be necessary and wise to create individual file systems and allocate a specific amount of storage to that file system using the EXT file system type. For example, if you create a separate file system for your database with the necessary disk space, you can prevent a problem of file system being filled up by other applications. In this case, only DB2 would have access to write data to the file system.

Example file system mount points or partitions are as follows:

```
/
/usr
/opt
/home
/var
swap
```

See Table 3-4 on page 38 for the actual file system sizes used during installation. Table 3-4 provides the basic set of guidelines for space required by each component. The values in this table were taken from our installed system after the installation using the file system layout values in Table 3-4 on page 38 and using the **df** and **du -ks** commands on each file system/product directory.

Note: There are many configuration scenarios to consider if you plan on using your system for Linux and Windows operating systems. If you plan to boot multiple operating systems, we recommend that you search for the latest information available on the Web, such as:

http://www.redhat.com/

The main issues to consider are related to the location of the Linux /boot file system being within the first 1024 cylinders, the use of GRUB or LILO (Windows 2000 requires its own boot manager), and the master boot record (MBR).

Table 3-4 Component disk space minimum requirements

Component	**File system (partitions)**	**Total disk required (MB)**[a]
Red Hat Linux 7.2	/ /boot /tmp /var /home /usr	954 (with the options selected)
	swap	1024 (1 GB RAM system)
IBM DB2 V7.2 Enterprise Edition	/usr /home	340

Component	File system (partitions)	Total disk required (MB)[a]
IBM HTTP Server V1.3.19.3	/opt	20
WebSphere Application Server V4.0.4 IBM Java 2 1.3	/opt	215 55
IBM WebSphere Commerce V5.4.0.3 Business Edition	/opt	1800
IBM WebSphere Commerce Payments V3.1.3	/opt	90

[a] These values are the minimum required to install the packages we chose to install. Your values may vary depending on selected packages and expected growth rate. If you have the disk space, you should consider being generous with your allocations.

3.2.4 Linux software packages

During the installation of Red Hat Linux, we chose not to install the Apache version that comes with the Linux media.

3.3 DB2 Server installation

This section provides detailed instructions for installing, configuring and verifying IBM DB2 Universal Database V7.2.1, Enterprise Edition for Linux plus IBM DB2 UDB V7 FixPak 7.

The section is organized into the following tasks:

- Pre-installation for DB2
- Install the DB2 Server
- Verify the DB2 Server installation
- Install DB2 UDB V7 FixPak 7
- Configure the DB2 Server
- Create the WebSphere Application Server repository database
- Stop DB2
- Start DB2

3.3.1 Pre-installation for DB2

Prior to installing DB2, ensure the following prerequisite file sets have been installed and tasks have been completed.

Prerequisite Linux packages for DB2

DB2 requires the packages listed in Table 3-5. Red Hat Linux 7.2 includes higher levels of these prerequisite packages. We used the higher levels supplied with Red Hat Linux 7.2. The package pdksh provides korn shell support needed by DB2 scripts. The ncurses package is used by the DB2 installer.

Table 3-5 DB2 Linux package prequisites

DB2 required package level	Red Hat Linux 7.2 package level used in the ITSO test environment
pdksh-5.2.14-12	pdksh-5.2.14-13
ncurses4-5.0-2	ncurses-5.2-12

The DB2 product documentation should be consulted for the official list of prerequisite software.

Note: For more detailed information on DB2 V7.2 and Linux, refer to:

http://www.linux.org/docs/ldp/howto/DB2-HOWTO/kernel24.html

To see if you have the packages installed on your system (listed in Table 3-5 on page 40), perform the following steps:

1. Verify whether the package is installed on the system using:

   ```
   # rpm --verify ncurses-5.2-12
   ```

   If no output is generated then the package is installed correctly; otherwise, the rpm package manager will return an error similar to the following:

   ```
   package ncurses-5.2-12 is not installed
   ```

2. Repeat step 1 for the package pdksh-5.2.14-13.
3. If the package is not installed, then you must locate and install it before proceeding with the DB2 installation. There are a number of ways to do this. The easiest way to install the package is if you have the original Red Hat Linux CD handy; otherwise, you can locate and download it from the Web.
4. The RPMs can be found on the Red Hat Linux 7.2 CDs as follows:
   - pdksh-5.2.14-13 can be found on CD2 in the /RedHat/RPMs directory.
   - ncurses-5.2-12 can be found on CD1 in the /RedHat/RPMs directory.

Note: The RPMs can also be obtained as follows:

Find the RPM on the Web:

- Go to the following site:
  http://rpmfind.net/
- Enter `ncurses4` into the search box and click **Search**, then download the RPM.

Find and download nucurses4 rpm for Red Hat Linux 7.2.

- Alternatively, you can find it in the directories of the Red Hat Linux FTP site:

  ftp://ftp.redhat.com

5. Once you have the RPM file either mounted or saved somewhere on your system, use the following commands to install or upgrade the required packages:

```
# rpm -U --nodeps /mnt/cdrom/RedHat/RPMS/ncurses-5.2-12.i386.rpm
# rpm -U --nodeps /mnt/cdrom/RedHat/RPMS/pdksh-5.2.14-13.i386.rpm
```

Pre-installation tasks

Prior to installing IBM DB2 Universal Database V7.2, Enterprise Edition, the following checks need to be completed:

- Verify that there are no existing active services that use the same DB2 TCP/IP ports on the server:
  - 523 (DB2 Server)
  - 50000 (default DB2 instance connection port)
  - 50001 (default DB2 instance interrupt port)
  - 50002 (DB2 Control Server)

  We suggest using the following command for this task:

  ```
  netstat -an | grep LISTEN
  ```

- If you will be running DB2 with multiple instances, you will need to increase the IPC limits on the database server:
  - Increase those immediately on the command line by running *one* of the following two commands:

    ```
    # sysctl -w kernel.msgmni=128
    ```

    or

    ```
    # echo “128” > /proc/sys/kernel/msgmni
    ```

- To increase the limits for each server reboot, add the lines listed in Example 3-1 to the end of the /etc/sysctl.conf file (or add one of the above two commands to your favorite startup file).

Example 3-1 IPC startup parameters for /etc/sysctl.conf file

```
# Sets maximum number of message queues to 128
# Set this to 1024 or higher on production systems
kernel.msgmni = 128
```

3.3.2 Install the DB2 Server

In order to install IBM DB2 Universal Database V7.2, Enterprise Edition for Linux, perform the following steps:

1. Log in as root and start a terminal session.
2. Mount the DB2 V7.2 CD-ROM:

   ```
   # mount /mnt/cdrom
   ```

Note: If you are running X Windows with either the KDE or Gnome window managers, the CD-ROM may automatically be mounted for you. To verify this, simply do a **`mount | grep /mnt/cdrom`** and if you get any output then the CD-ROM has already been mounted.

3. Start the DB2 installer program:

   ```
   # cd /mnt/cdrom
   # ./db2setup
   ```

   Note: To generate a trace log, enter the **`./db2setup -d`** command instead of the **`./db2setup`** command. The **`./db2setup -d`** command generates a log in /tmp/db2setup.trc.

4. In the Install DB2 V7.2 window, select only the following options:
   - DB2 UDB Enterprise Edition
   - DB2 Application Development Client

     This option is required to load stored procedures needed for WebSphere Commerce.

Tip: Navigation tips:

- Press **Tab** to move between available options and fields.
- Press **Ctrl+L** to refresh the window at any point.
- Select and press **Enter** to select an option.

5. Select the DB2 Product Library's **Customize** option and press **Enter**.
6. In the DB2 Product Library window, highlight the appropriate option for your locale under the DB2 Product Library (HTML) section, then highlight **OK** and press **Enter**.
7. Select **OK** and press **Enter**.
8. In the Create DB2 Services window, select the **Create a DB2 Instance** option, highlight **OK** and press **Enter**.
9. When the DB2 instance authentication window appears, enter the following:
   - User Name: `<db2_instance_owner>`

     For example, `db2inst1`
   - User ID: `<use default UID>`
   - Group Name: `db2iadm1`
   - Group ID: `<use default GID>`
   - Home Directory: `/home/<db2_instance_owner>`

     For example, `/home/db2inst1`
   - Password: `<user_password>`
   - Verify Password: `<user_password>`

Important: The DB2 installer uses the above information to automatically perform the following operations:

- Create a group db2iadm1
- Create a user <db2_instance_owner> with primary group db2iadm1
- Set the <db2_instance_owner> password to <user_password> value. The password used must meet DB2 requirements: eight characters or fewer and not containing the characters "<" or ">".
- Change the ownership (owner:group) of the /home/<db2_instance_owner> directory to be <db2_instance_owner>:db2iadm1

10. Select **OK** and press **Enter**.

11. In the Fence User window, enter the following:

   – User Name: `db2fenc1`
   – User ID: `<use default UID>`
   – Group Name: `db2fadm1`
   – Group ID: `<use_default_GID>`
   – Home Directory: `/home/db2fenc1`
   – Password: `<db2fenc1_password>`
   – Verify password: `<db2fenc1_password>`

Important: The DB2 installer uses the above information to automatically perform the following operations:

- Create a group db2fadm1
- Create a user db2fenc1 with primary group db2fadm1
- Set the db2fenc1 password to <db2fenc1_password> value. The password used must meet DB2 requirements: eight characters or fewer and not containing the characters “<“ or “>”.
- Change the ownership (owner:group) of the /home/db2fenc1 directory to be db2fenc1:db2fadm1

12. Select **OK** and press **Enter**.
13. In the DB2 Warehouse Control Database window, select **Do not setup DB2 Warehouse Control Database**, then highlight **OK**, and press **Enter**.
14. In the Create DB2 Services window, highlight the **Create Administration Server** option (*optional step*), then enter the following:

   – User Name: `db2as`
   – User ID: `<use default UID>`
   – Group Name: `db2asgrp`
   – Group ID: `<use default GID>`
   – Home Directory: `/home/db2asgrp`
   – Password: `<db2asgrp_password>`
   – Verify password: `<db2asgrp_password>`

Important: The DB2 installer uses the above information to automatically perform the following operations:

- Create a group db2asgrp
- Create a user db2as with primary group db2asgrp
- Set the db2as password to <db2as_password> value. The password used must meet DB2 requirements: eight characters or fewer and not containing the characters “<“ or “>”.
- Change the ownership (owner:group) of the /home/db2as directory to be db2as:db2asgrp

15. Select **OK** and press **Enter**.
16. A message window appears indicating that DB2SYSTEM will be set to <hostname>. Select **OK** and press **Enter**.
17. Back in the Create DB2 Services window, highlight **OK** and press **Enter**.
18. The Summary Report window is displayed, listing the product components to be installed. Select **Continue** and press **Enter**.
19. A warning window appears indicating `This is your last chance to stop`. Select **OK** and press **Enter**.
20. The db2setup program installs the selected components. Depending on the speed of your processor, this can take up to 15 minutes.
21. You may be prompted to register the product. Complete the registration, then exit back to the install window.
22. When the install completes, a notice window informs you of whether the installation was successful. Select **OK** and press **Enter**.
23. Scan the Status Report to ensure that all components were installed successfully. Select **OK** and press **Enter**.
24. In the DB2 Installer window, highlight **Close** and press **Enter**.
25. A window appears asking `Do you want to exit the DB2 Installer?` Select **OK** and press **Enter**.
26. Unmount the CD-ROM:

    ```
    cd /
    umount /mnt/cdrom
    ```

The DB2 installation is now complete.

3.3.3 Verify the DB2 Server installation

To verify the DB2 Server installation, complete the following tasks:

- Verify home directory permissions
- Verify the DB2 instance owner profile
- Verify the DB2 instance symbolic links
- Verify the DB2 release level
- Verify the DB2 service name
- Verify the database manager configuration
- Create the DB2 sample database

Verify home directory permissions

Check that the home directory ownership has been correctly set up by the db2setup program:

Table 3-6 DB2 home directory required permissions

Home directory path	Owner	Group	Permissions[a]
/home/<db2_instance_owner>	<db2_instance_owner>	db2iadm1	drwxr-xr-x
/home/db2fenc1	db2fenc1	db2fadm1	drwxr-xr-x
/home/db2as	db2as	db2asgrp	drwxr-xr-x

[a] Permissions needed so that the DB2 instance owner can read, write, and execute files and directories within the path. Group members' and other users' access rights are up to each company's security policies and business needs.

If a DB2 related home directory has not been correctly configured, perform the following steps:

1. Log in as root and start a terminal session.
2. Issue the following command, substituting values from the table above:

```
# chown -fR <owner>:<group> <home_directory_path>
```

Verify the DB2 instance owner profile

The DB2 Server installation should set up the .bashrc environment file of the <db2_instance_owner> (found in root of home directory) so that the DB2 environment is set up when the user logs in.

1. Log in as user <db2_instance_owner>. For example:

```
# su - db2inst1
```

2. The following content should have been added to the .bashrc file:

   ```
   if [ -f /home/db2inst1/sqllib/db2profile ] ; then
   . /home/db2inst1/sqllib/db2profile
   fi
   ```

3. If not present, manually edit the file to add the above content.

Verify the DB2 instance symbolic links

The DB2 Server installation automatically creates a DB2 instance <db2_instance_owner> under the /home/<db2_instance_owner> directory. As part of the instance creation, db2setup should create symbolic links in the /home/<db2_instance_owner>/sqllib directory to files under /usr/IBMdb2/V7.1.

Perform the following steps to check whether the symbolic links have been created:

1. Log in as root, and start a terminal session.
2. Change directory to /home/<db2_instance_owner>/sqllib.
3. Check whether a number of symbolic links exist pointing to files under /usr/IBMdb2/V7.1.
4. If not, issue the following commands:

   ```
   # cd /usr/IBMdb2/V7.1/cfg
   # ./db2ln
   ```

Verify the DB2 release level

Check that DB2 has the correct internal release level to meet WebSphere Application Server requirements:

1. Change to user <db2_instance_owner>. For example:

   ```
   # su - db2inst1
   ```

2. Enter the following command:

   ```
   $ db2level
   ```

Verify the DB2 service name

1. Open the /etc/services file and locate the entries that have comments referring to the DB2 instance connection port.
2. Locate the service name in the first column that corresponds to the lower port number. For example, if the following services were displayed:

   ```
   db2cdb2inst1 50000/tcp #Connection port for DB2 instance db2inst1
   db2idb2inst1 50001/tcp #Interrupt port for DB2 instance db2inst1
   ```

   then record the db2cdb2inst1 service name for later use.

Verify the database manager configuration

Check the service name is recorded in the database manager configuration:

1. Change to user <db2_instance_owner>:

   ```
   # su - <db2_instance_owner>
   ```

2. Enter the following command:

   ```
   $ db2 get dbm cfg | grep SVCENAME
   ```

3. Verify that the SVCENAME value matches the service name recorded above from the `services` file. For example, something similar to the following should be displayed:

   ```
   TCP/IP Service name (SVCENAME)=db2cdb2inst1
   ```

4. If the value does not match, update the database manager configuration using the following commands:

   ```
   $ db2 update dbm cfg using svcename db2cd
   $ db2stop
   $ db2start
   ```

   Where the value of `SVCENAME` must be replaced with the service name.

Create the DB2 sample database

The DB2 installation can be tested by creating and connecting to the sample database supplied with DB2 specifically for this purpose:

1. Change to user <db2_instance_owner>. For example:

   ```
   # su - db2inst1
   ```

2. Create the DB2-supplied sample database, named sample, to verify that DB2 is working properly by typing the following command:

   ```
   $ db2sampl
   ```

3. List all DB2 databases for the DB2 instance:

   ```
   $ db2 list db directory
   ```

   This should give output containing the following:

   ```
   Database 1 entry:
    Database alias                  = SAMPLE
    Database name                   = SAMPLE
    Database drive                  = /home/db2inst1
    Database release level          = 9.00
    Comment                         =
    Directory entry type            = Indirect
    Catalog node number             = 0
   ```

4. Test the connectivity to the database:

   ```
   $ db2 connect to sample
   $ db2 disconnect current
   ```

5. Delete the test database.

   ```
   $ db2 drop db sample
   ```

3.3.4 Install DB2 UDB V7 FixPak 7

IBM DB2 Universal Database V7 FixPak 6 is required for IBM WebSphere Commerce V5.4 for Linux. In our example, we used the most recent version of DB2 FixPak at the time of writing, DB2 UDB V7 FixPak 7 (FP7_U484487.tar 9/9/2002). Refer to the FixpakReadme.txt for details.

> **Note:** During the development of this Redpaper, we noticed that there were at least two versions of DB2 UDB V7 FixPak 7 made availabe on the Web. Ensure that you have the latest version. We used the DB2 UDB V7 FixPak 7 dated 9/9/2002.

The high level steps to install DB2 UDB V7 FixPak 7 are as follows (refer to FixpakReadme.txt file for detail installation information):

1. This FixPak can only be applied to systems running DB2 UDB V7.2 or DB2 UDB V7.1 with FixPak 3 (level 7.1.0.40) or higher. To determine the current level of DB2 UDB which has been installed, run the following command:

   ```
   # rpm -qa | grep db2
   ```

   The output of the command should include the following:

   ```
   db2cliv71-7.1.0-x
   ```

   Where `x` is the fix level (40 or higher).

2. Before installing the FixPak, ensure that all DB2 processes in all DB2 instances have been stopped. This includes the DB2 Administration Server.

   ```
   # su - <db2_instance_owner>
   $ . $HOME/sqllib/db2profile
   $ db2 force applications all
   $ db2 terminate
   $ db2stop
   $ db2licd end      # run at each physical node
   $ exit
   ```

   Where `iname` is the instance owner name.

Then, while still logged on as root, enter the following commands:

```
# su - <aname>
$ . $HOME/sqllib/db2profile
$ db2admin stop
$ exit
```

Where `aname` is the administration server name.

3. DB2 UDB V7 FixPak 7 can be downloaded from:

   ```
   ftp://ftp.software.ibm.com/ps/products/db2/fixes/english-us/db2linuxv7/
   FP7_U484487/
   ```

4. As the root user, decompress the FixPak file to a temporary directory on the system (for example, /tmp). You will need over 330 MB of free disk space in that directory. The following example command will directly decompress the tar file on the CD and extract it to the current directory, in a single step:

   ```
   # cd /tmp
   # tar -xvf FP7_U484487.tar
   ```

5. The tar file will be extracted into the delta_install subdirectory. Install the FixPak enter the following command:

   ```
   # su - root
   # cd FP7_U484487
   # ./installFixPak
   ```

6. After installation, update DB2 instance by entering the following command:

   ```
   # /usr/IBMdb2/V7.1/instance/db2iupdt db2inst1
   ```

7. Restart DB2:

   ```
   # su - db2inst1
   $ db2start
   ```

Note: If you have an existing database, you will need to complete the binding procedures found in the FixpackReadme file.

3.3.5 Configure the DB2 Server

After the DB2 Server installation, many configuration tasks must be performed in preparation for the WebSphere Application Server installation.

This section is organized into the following tasks:

- Update root administrative groups
- Update JDBC level
- Configure the TCP/IP communication mode
- Verify the DB2 environment
- Update the root environment file

Update root administrative groups

The DB2 Server installation should add the db2asgrp administrative group to the root user.

Perform the following steps to check whether the root account's administrative groups have been amended:

1. Log in as root, and start a terminal session.
2. Issue the following command:

   ```
   # groups
   ```

3. If db2asgrp is not listed as one of the groups assigned to root, you can use the Red Hat GUI tools to reconfigure the root user or you can run the following:

   ```
   # vigr
   ```

   This will bring you into a vi editing session with a locked version of the /etc/group file. Find the line in the file that starts with `db2asgrp:` and add the user, root, to the end of it (separated by a comma from other user names). The line should look similar to this after editing:

   ```
   db2asgrp:db2as,root
   ```

   Once you have added that, press **Esc+:wq** to save and quit vi. The system will then prompt you to edit the shadow group file. Choose **Yes** and do the same thing to that file, saving and quitting at the end of your editing.

Update JDBC level

The WebSphere Application Server V4.0.1 requires the use of JDBC 2.0, whereas the default installation of IBM DB2 V7.2 uses JDBC 1.2. To update the DB2 JDBC level, complete the following steps:

1. Change to user <db2_instance_owner>:

   ```
   # su - <db2_instance_owner>
   ```

2. Add the following content to the end of the <db2_instance_owner> .bashrc environment file:

   ```
   if [ -f ~/sqllib/java12/usejdbc2 ] ; then
   . ~/sqllib/java12/usejdbc2
   fi
   ```

Update msgmni

If you will be running DB2 with multiple instances or many databases within an instance, you will need to increase the IPC limits on the database server:

- Increase it immediately on the command line by running *one* of the following two commands:

  ```
  sysctl -w kernel.msgmni=128
  ```

  or

  ```
  echo "128" > /proc/sys/kernel/msgmni
  ```

- Also, increase it for each server reboot by adding the lines in Example 3-2 to the end of the /etc/sysctl.conf file (or adding one of the above two commands to your favorite startup file):

Example 3-2 IPC startup parameters for /etc/sysctl.conf file

```
# Sets maximum number of message queues to 128
# Set this to 1024 or higher on production systems
kernel.msgmni = 128
```

Configure the TCP/IP communication mode

The DB2 Server may need to be re-configured to use TCP/IP as its primary communication method by completing the following steps:

1. Change to user <db2_instance_owner>:

   ```
   # su - <db2_instance_owner>
   ```

2. Check whether TCP/IP is the current DB2 communication method. The following command should return a value of `tcpip`:

   ```
   $ db2set DB2COMM
   ```

3. If not, reset the DB2COMM DB2 environment variable:

   ```
   $ db2set DB2COMM=tcpip
   ```

Verify the DB2 environment

After the above configuration steps, we need to check that the environment being set up by the db2profile and usejdbc2 scripts is correct:

1. Change to user <db2_instance_owner> (for example, db2inst1).

   ```
   # su - db2inst1
   ```

2. Issue the following command:

   ```
   $ set | grep [Dd][Bb]2
   ```

3. Verify that the environment variables in this output match the values in Table 3-7, where the DB2 instance owner is db2inst1.

Table 3-7 DB2 Server required environment variables

Environment variable	Required value
CLASSPATH	/home/db2inst1/sqllib/function:/home/db2inst1/sqllib/java12/db2java.zip:/home/db2inst1/sqllib/java/runtime.zip
DB2DIR	/usr/IBMdb2/V7.1
DB2INSTANCE	db2inst1
INSTHOME	/home/db2inst1
LD_LIBRARY_PATH	:/home/db2inst1/sqllib/lib
LIBPATH	:/home/db2inst1/sqllib/lib
PATH	.......:/home/db2inst1/sqllib/bin:/home/db2inst1/sqllib/adm:/home/db2inst1/sqllib/misc

Update the root environment file

By default, the WebSphere Application Server will be run under root and will require access to the DB2 environment so that it can access the WebSphere Application Server administration database. In our example, after installing WebSphere Commerce, we will run the Application Server as a non root user.

1. This requires that the root account's environment .bashrc file, found in the /root directory, be edited to add the content as seen in Example 3-3 at the end of the .bashrc file.

Example 3-3 Sample entries in the root .bashrc file for DB2

```
# Setup DB2 environment for root user.
if [ -f /home/db2inst1/sqllib/db2profile ] ; then
. /home/db2inst1/sqllib/db2profile
fi

# Force DB2 to use JDBC 2.0.
if [ -f /home/db2inst1/sqllib/java12/usejdbc2 ] ; then
. /home/db2inst1/sqllib/java12/usejdbc2
fi
```

where `db2inst1` is the DB2 instance owner in Example 3-3.

2. Log out and log back in for the changes to take effect.

3.3.6 Create the WebSphere Application Server repository database

The following steps create the WebSphere Application Server repository database, also known as the WebSphere Application Server database. The database will be populated with WebSphere Application Server schema and default values in a later task.

To set up the WebSphere Application Server database, complete the following steps:

1. Log in as <db2_instance_owner>.

   ```
   # su - db2inst1
   ```

2. Create the WebSphere Application Server repository database.

   ```
   $ db2 create db was4
   ```

   Note: Although the repository database created here is called was4, any valid DB2 database name can be used.

3. Configure the heap size of the WebSphere Application Server repository database as required by the WebSphere Application Server:

   ```
   $ db2 update db cfg for was4 using applheapsz 512
   ```

4. Verify that the new database is known to DB2:

   ```
   $ db2 list db directory
   ```

   You should see a message containing the following output:

   ```
   Database 1 entry:
    Database alias                  = WAS4
    Database name                   = WAS4
    Database drive                  = /home/db2inst1
    Database release level          = 9.00
    Comment                         =
    Directory entry type            = Indirect
    Catalog node number             = 0
   ```

5. Catalog the TCP/IP node.

   In order to access the administration database via TCP/IP, the DB2 node must first be catalogued.

Syntax:

```
db2 catalog tcpip node <node_name> remote <local_hostname> server
<service_name>
```

The `<service_name>` used to catalog the node must be the same as the database instance connection port name in the /etc/services file. The `<node_name>` chosen can be any valid DB2 node name.

For example, when we configured the DB2 server on the same node (comlx1) as the application server for the single-tier runtime, we entered the following:

```
$ db2 catalog tcpip node comlx1 remote comlx1 server db2cdb2inst1
```

6. Verify the attach to the TCP/IP node.

Syntax:

```
db2 attach to <node_name> user <db2_instance_owner> using
<db2_instance_owner_passwd>
```

For example, when we configured the DB2 server on the same node (comlx1) as the application server for the single-tier runtime, we entered the following:

```
$ db2 attach to comlx1 user db2inst1 using <db2inst1_password>
```

7. Catalog the database.

   The administration database must now be catalogued as part of this TCP/IP node.

Syntax:

```
db2 catalog db <database_name> as <database_alias> at node <node_name>
```

For example, when we configured the DB2 server on the same node (comlx1) as the application server for the single-tier runtime, we entered the following:

```
$ db2 catalog db was4 as was40 at node comlx1
```

8. Verify that the database alias is known to DB2:

```
$ db2 list db directory
```

The output should contain something like the following message:

```
Database 2 entry:
 Database alias                  = WAS40
 Database name                   = WAS4
 Node name                       = <node_name>
 Database release level          = 9.00
 Comment                         =
 Directory entry type            = Remote
 Catalog node number             = -1
```

9. Verify a connection to the local database via TCP/IP:

   ```
   $ db2 connect to was40 user db2inst1 using db2inst1
   $ db2 disconnect current
   ```

   Where `db2inst1` is the DB2 instance owner password on the local DB2 server.

   > **Tip:** When the DB2 administration account is used to create a new database, it is automatically granted DBA access rights. You only need to specifically grant access to another user if you plan to access the database from an account other than <db2_instance_owner>.

10. Reboot the system for the changes to take effect. *Alternatively*, you can restart the DB2 processes by entering the following:

    ```
    $ su - db2inst1 “-c db2stop”
    su - db2as “-c db2stop”
    su - db2as “-c db2start”
    su - db2inst1 “-c db2start”
    ```

3.3.7 Stop DB2

To stop DB2, enter the following:

```
# su - db2inst1
$ db2 force applications all
$ db2 terminate
$ db2stop
$ db2licd end      # run at each physical node
$ exit
```

3.3.8 Start DB2

To start DB2, enter the following:

```
# su - db2inst1
$ db2start
```

3.4 IBM HTTP Server and WebSphere plugin installation

The IBM HTTP Server V1.3.19 can be installed independently or as part of the WebSphere Application Server V4 installation. We choose to download and install IBM HTTP Server V1.3.19.3 separately, for the following reasons:

- Verify IBM HTTP Server SSL configuration

  We can verify the SSL prior to installing the WebSphere Application Server.

- Modular installation methodology

  A modular installation approach can be reused when adding a remote Web server, a shown in Chapter 5, “Implementing a 3-tier runtime with the addition of a remote Web server” on page 133.

- Avoid GSKit conflict

  We can avoid a GSKit installation conflict between IBM HTTP Server V1.3.19 supplied with WebSphere Application Server V4.0.1 and the WebSphere Application Server V4 FixPak 3.

Note: To avoid the GSKit installation conflict and resulting failure, we suggest one of the following installation workarounds:

- Install the IBM HTTP Server V1.3.19.3 downloaded from the Web. This procedure is documented in 3.4.1, “Install the IBM HTTP Server” on page 58.
- If you choose to install the IBM HTTP Server included with WebSphere Application Server V4.0.1, you will need to move the gsk5bas-5.xxxx.rpm found in the /opt/IBMHTTPServer directory to a temporary directory or rename the file.

  Also, verify that the /etc/ld.so.conf file containing the following entry for the GSKit lib path is available to the operating system:

  ```
  /usr/local/ibm/gsk5/lib
  ldconfig -v
  ```

  Alternatively, the GSKit lib path is symbolically linked in the /usr/lib directory.

This section includes the following tasks:

- Install the IBM HTTP Server
- Install the WebSphere plugin for the IBM HTTP Server
- Install the WebSphere Application Server V4 FixPak 4
- Cumulative WebSphere plugin e-fix (PQ63508)
- Configure the IBM HTTP Server
- Restart, stop and start IBM HTTP Server
- Verify the IBM HTTP Server

3.4.1 Install the IBM HTTP Server

This section describes how to download and install the IBM HTTP Server V1.3.19.3. This version of the IBM HTTP Server is needed to address a GSKit installation conflict.

To download and install the IBM HTTP Server, do the following:

1. Enter the following URL in a Web browser:

   `http://www6.software.ibm.com/dl/websphere/http-p`

   > **Note:** In order to download the compressed tar file, you need to be a registered user.

2. Download the HTTPServer.linux.128.EN.tar.gz file to the /tmp directory.
3. Decompress the download file as follows:

   ```
   # cd /tmp
   # tar -xvzf HTTPServer.linux.128.EN.tar.gz
   ```

   Alternatively, enter the following command:

   ```
   # gzip -d -c HTTPServer.linux.128.EN.tar.gz | tar -xvf -
   ```

4. Change to the installation directory created from uncompressing the zipped tar file:

   ```
   cd /tmp/IHS-1.3.19.3
   ```

5. You will need to install each of the RPMs listed in Table 3-8 in the order list to avoid dependency issues, by entering the following command:

   ```
   # rpm -U --nodeps <rpm_name>
   ```

   Where `<rpm_name>` is name of the RPM. Replace `<rpm_name>` with each of the RPM packages listed in the Table 3-8.

Note: The `rpm` command will not return any output if it was successful, but will if there are any problems. If greater verbosity is desired, the following command could be used instead:

```
rpm -Uvh --nodeps <rpm_Package>
```

Table 3-8 IBM HTTP Server V1.3.19.3 RPM packages

RPM packages	Description
gsk5bas-5.0-4.79.i386.rpm	IBM GSKit 5.
IBM_HTTP_Server-1.3.19-3.i386.rpm	IBM HTTP Server
IBM_ADMIN_EN-1.3.19-3.i386.rpm	IBM HTTP Server Administration Server English message files
IBM_ADMIN_Server-1.3.19-3.i386.rpm	IBM HTTP Server Administration Server
IBM_FastCGI-1.3.19-3.i386.rpm	Implementation of FastCGI standard - increases CGI performance. **Note**: This may be optional.
IBM_MAN_ENU-1.3.19-3.i386.rpm	IBM HTTP Server English language MAN pages for help
IBM_MSG_EN-1.3.19.3.i386.rpm	IBM HTTP Server English message files
IBM_SSL_128-1.3.19-3.i386.rpm	IBM HTTP Server 128 bit SSL support **Note**: Required for 128 bit SSL encryption for the US and selected countries. **Note**: Replace package with the appropriate 56 bit rpm if 56 bit encryption is used.
IBM_SSL_Base-1.3.19-3.i386.rpm	IBM HTTP Server base SSL support
IBM_SSL_EN-1.3.19-3.i386.rpm	IBM HTTP Server 128 bit SSL support message files

3.4.2 Install the WebSphere plugin for the IBM HTTP Server

The WebSphere plugin installation is required for the IBM HTTP Server to redirect requests to the WebSphere Application Server based on entries found in the plugin-cfg.xml file. In addition, when installing the WebSphere plugin, the IBM JDK is installed, which contains the JRE. The JRE is needed by the Java-based IKeyman application provided with the IBM HTTP Server to create the key database and add an SSL certificate.

To install the WebSphere plugin, complete the following steps:

1. Log in as root and start a console session.
2. Ensure the DISPLAY and TERM environment variables are properly set.
3. Insert the IBM WebSphere Application Server V4.0.1 CD ROM into the CD-ROM drive and mount the CD. By default, Red Hat Linux will mount the CD as /mnt/cdrom. Alternatively, mount the CD ROM as follows:

   ```
   # mount -r /dev/cdrom /mnt/cdrom
   ```

4. Change to the installation directory:

   ```
   # cd /mnt/cdrom
   ```

5. Run the install.sh installation script:

   ```
   # ./install.sh
   ```

6. In the Welcome window, click **Next**.

> **Important:** If prerequisite checking is enabled for UNIX, then an alert may be displayed indicating that some of the installation prerequisites have not been met or higher levels have been found. This can even occur if newer patches/packages than those listed in prereq.properties are installed. If such an alert is displayed, recheck all prerequisites, and if they are met or exceeded, perform the following steps:
>
> - Exit the installation.
> - Disable prerequisite checking in the prereq.properties file.
> - Restart the installation from the beginning.

7. In the Installation Options window, select **Custom Installation** and click **Next**.
8. In the Choose Application Server Components window, select only **Web server Plugins**, and then click **Next**.

   By default, all options are selected. Deselect all options but the Web server Plugins option.

9. Select the **IBM HTTP Server Plugin**, then click **Next**.
10. When the Prerequisite Check window appears, you may see a message stating that a higher version was found. This is because we have installed the IBM HTTP Server V1.3.19.3, which is a higher level than the prereq. Click **OK** to continue.
11. Accept the default destination directory (/opt/WebSphere/AppServer), and click **Next**.
12. Review the options selected and then click **Install** to begin the installation.

13. When prompted to enter the location of the `httpd.conf` file, do the following:

    a. Open a new Linux Console window.

    b. Change to the `/opt/IBMHTTPServer/conf` directory.

    c. Copy httpd.conf to httpd.conf.org.

    ```
    # cp httpd.conf httpd.conf.org
    ```

    d. Copy httpd.conf.sample to httpd.conf.

    ```
    # cp httpd.conf.sample httpd.conf
    ```

14. After copying the httpd.conf.sample to httpd.conf file, enter the following path and filename in the Installer window then click **Next**:

    ```
    /opt/IBMHTTPServer/conf/httpd.conf
    ```

15. When the Setup Complete window appears, click **Finish**.

3.4.3 Install the WebSphere Application Server V4 FixPak 4

For this scenario, we do *not* need to service the *IBM HTTP Server*, but do want to service the WebSphere plugin for the IBM HTTP Server and the IBM JDK. Please pay special attention to the prompts during the FixPak 4 installation.

For detailed information on the FixPak installation, refer to 3.5.3, “Install WebSphere Application Server V4 FixPak 4” on page 73 for details. When installing WebSphere Application Server V4 FixPak 4 for this scenario, take note of the following requirements:

1. Ensure the WebSphere Application Server is stopped (it should not be installed yet if you are following the procedure in this chapter).
2. When prompted, do *not* install service for the *IBM HTTP Server* (following our procedure, we have already installed a higher service level of the IBM HTTP Server 1.3.19.3).
3. When prompted, update WebSphere Application Server and the JDK.

3.4.4 Cumulative WebSphere plugin e-fix (PQ63508)

After installing WebSphere Application Server V4 FixPak 4, download and install the WebSphere plugin e-fix PQ63508 to avoid AdminConsole AppServer environment variables being overwritten when updating other properties. The e-fix can be downloaded at:

```
http://www-1.ibm.com/support/docview.wss?rs=180&context=SSEQTP&uid=swg24001
801
```

Refer to the readme for details.

3.4.5 Configure the IBM HTTP Server

This section describes how to configure the IBM HTTP Server, including the following tasks:

- Create the HTTP Server admin account
- Create an HTTP Administration Server runtime user account
- Create an HTTP Server runtime user account
- Configure the HTTP Server httpd.conf for SSL
- Create a key database for the IBM HTTP Server
- Create a certificate for the IBM HTTP Server
- Configuring the ServerName
- Restart the IBM HTTP Server
- Stop the IBM HTTP Server
- Start the IBM HTTP Server

Create the HTTP Server admin account

If you should decide to run the HTTP Administration Server, the instructions in this section should be followed. Otherwise, this section can be skipped.

The admin account is used to access the IBM HTTP Administration Server Configuration GUI. To create the account, perform the following steps:

1. Log in as root, and start a terminal session.
2. Change directory to the <http_home>/bin directory. For example:

   ```
   # cd /opt/IBMHTTPServer/bin
   ```

3. Create HTTP admin account <http_admin_account>. For example:

   ```
   # ./htpasswd -m ../conf/admin.passwd httpadm
   ```

   Where `httpadm` is the admin user account. The **`htpasswd`** command will prompt you for a new password,and then for you to retype the new password.

Syntax: Create HTTP admin account:

```
./htpasswd -m ../conf/admin.passwd <http_admin_user>
```

Create an HTTP Administration Server runtime user account

If you should decide to run the HTTP Administration Server the instructions in this section should be followed. Otherwise this section can be skipped.

Although the Admin HTTP Server process is started under the root account, it must be configured to then switch and run under another account. A UNIX account can be created specifically for the purpose.

Perform the following steps to create the UNIX account and configure the Admin HTTP Server:

1. Log in as root, and start a terminal session.
2. Change directory to the <http_home>/bin directory. For example:

   ```
   # cd /opt/IBMHTTPServer/bin
   ```

3. Run the setupadm script:

   ```
   # ./setupadm
   ```

   Answer the prompts as follows:

   a. Supply a user ID to run the Administration Server. For example, `httprun`

   b. Supply a Group Name to run the Administration Server. For example, `httpgrp`

   c. Supply the directory containing the files for which a change in the permissions is necessary. The default is /opt/IBMHTTPServer/conf.

      Press **Enter** to accept the default.

   d. To perform the change, enter `1`. To quit with no changes, enter `2` (default).

      Type `1` and press **Enter** to perform the changes.

   e. The configuration file /opt/IBMHTTPServer/conf/admin.conf will be saved. If you wish to update the Administration Server Configuration file, enter `1`. To exit with no change, enter `2` (default).

      Type `1` and press **Enter** to update configuration file.

   f. If you wish to run the Admin Server and IHS Server in a language other than English, enter `1`. For English, enter `2` (default).

      Press **Enter** to accept the default (English).

4. The setupadm program returns to the system prompt.

Create an HTTP Server runtime user account

This section is optional. By default, the configuration file (httpd.conf) is set up to run the HTTP Server as user *nobody* and group *nobody*. This should be sufficient for most installations; however, some companies prefer not to do this and, instead, create separate Web server accounts. If the latter is the case for you, then you will need to first create the desired user and group on the Linux system using a command such as the following. Otherwise, the user/group setting in the httpd.conf file can be left at defaults:

```
# useradd www
```

This will also create a group called www automatically.

Edit the <http_home>/conf/httpd.conf file and update the following settings.

Table 3-9 httpd.conf required settings

Setting	Required value...
User	www
Group	www

Configure the HTTP Server httpd.conf for SSL

Enabling SSL support for the IBM HTTP Server is often required in a production Web server runtime environment. To configure the IBM HTTP Server with SSL support, complete the following steps:

1. Stop the IBM HTTP Server:

   ```
   # cd /opt/IBMHTTPServer/bin
   # ./apachectl stop
   ```

2. Change to the conf directory:

   ```
   # cd /opt/IBMHTTPServer/conf
   ```

3. Modify the httpd.conf file with a text editor (for example, vi). Uncomment the following lines by removing the # sign:

   ```
   LoadModule ibm_ssl_module libexec/mod_ibm_ssl_<encryption_level>.so
   ```

   Where the `<encryption_level>` is the appropriate level for your locale (for example, 128 for en_US).

   Remove the comment character from the following lines to configure the SSL module in the HTTP Server:

   ```
   AddModule mod_ibm_ssl.c
   Listen 80
   Listen 443
   <VirtualHost host.domain.com:443>
   ```

   Note: You must substitute your fully qualified host name in this line (for example, `<VirtualHost server.itso.ral.ibm.com:443>`).

   Uncomment the following:

   ```
   SSLEnable
   </VirtualHost>
   SSLDisable
   Keyfile "/opt/HTTPServer/keys/http_key.kdb"
   ```

   Note: Replace the word `keys` with `ssl`.

Uncomment the SSL timeout values:

```
SSLV2Timeout 100
SSLV3Timeout 1000
```

4. Save your changes.

Create a key database for the IBM HTTP Server

To create a key database to manage certificates for the IBM HTTP Server, complete the following steps:

1. Ensure the JRE 1.3 or higher is installed to execute IKeyman.

   The JRE is installed as part of the JDK install done automatically when installing the WebSphere plugin (refer to 3.4.2, “Install the WebSphere plugin for the IBM HTTP Server” on page 59).

2. Set the Java path by entering the following:

   ```
   # cd /opt/IBMHTTPServer/ssl
   # export JAVA_HOME=/opt/WebSphere/AppServer/java
   ```

3. Start IBM Key Management utility by typing the following command:

   ```
   # ikeyman
   ```

4. When the IBM Key Management window appears, click the **Key Database File** menu and select **New.**

5. When the New window appears, enter the following and then click **OK**:

   – File name: `http_key.kdb`

   – File location: `/opt/HTTPServer/ssl`

     The path will be entered by default if you started the IKeyman utility from this directory.

6. When the Password Prompt window appears, enter your password and then click **OK**. This password is used to access the keyfile.

   – Select **Set expiration time** and change it to the desired number of days (for example, 360).

   – Enable **Stash the password to a file**.

7. When the Information window appears, confirming the password has been saved and encrypted in the keyfile.sth file, click **OK.**

Create a certificate for the IBM HTTP Server

To create a self-signed certificate for the IBM HTTP Server, complete the following steps:

> **Note:** For production SSL enablement, you will need to get a real SSL certificate from an organization such as VeriSign at:
>
> `http://www.verisign.com/`
>
> For the purposes of this Redpaper and testing, we created a self-signed certificate.

1. From the IBM Key Management utility, click **Create** and select **New Self-Signed Certificate.**
2. When the Create New Self-Signed Certificate window appears, enter the following (required fields) and then click **OK**.
   - Key Label: `http_ssl` (user-defined name)
   - Organization: `IBM` (your company name)
3. Close the IBM Key Management utility.

> **Note:** The IBM Key Management Utility program may have a problem closing. If this happens, use the Linux **`kill app`** option to stop the utility.

4. Changes will not take effect until the HTTP Server is restarted.

Configuring the ServerName

To configure the IBM HTTP Server ServerName to include a fully qualified hostname, complete the following steps:

1. Stop the IBM HTTP Server:

   ```
   # cd /opt/IBMHTTPServer/bin
   # ./apachectl stop
   ```

2. Change to the directory of the httpd.conf:

   ```
   # cd /opt/IBMHTTPServer/conf
   ```

3. Modify the httpd.conf file to update the value of the ServerName keyword as the fully qualified hostname.

   For example:

   ```
   ServerName comlx1.itso.ral.ibm.com
   ```

4. Start the HTTP Server:

   ```
   # cd /opt/IBMHTTPServer/bin
   # ./apachectl start
   ```

3.4.6 Restart, stop and start IBM HTTP Server

The IBM HTTP Server can be restarted, stopped and started as follows.

Restart the IBM HTTP Server

To restart the IBM HTTP Server (Apache) style startup, log in as root and type the following command:

```
/opt/IBMHTTPServer/bin/apachectl restart
```

For a Red Hat specific startup, log in as root and type the following command:

```
/etc/rc.d/init.d/ibmhttpd restart
```

Stop the IBM HTTP Server

To stop the IBM HTTP Server, enter the following:

```
/opt/IBMHTTPServer/bin/apachectl stop
```

Start the IBM HTTP Server

To start the IBM HTTP Server, enter the following:

```
/opt/IBMHTTPServer/bin/apachectl start
```

3.4.7 Verify the IBM HTTP Server

In order to verify the IBM HTTP Server installation, perform the following checks on the Web server machine:

- Check process status
- Check request handling

Check process status

1. Check that the HTTP Server processes are running by issuing the following command:

   ```
   ps -ef | grep httpd
   ```

   The output should list a number of httpd processes. The output should look similar to the output in Figure 3-1.

```
root     27585     1  0 Aug30 ?        00:00:00 /opt/IBMHTTPServer/bin/httpd
www      27586 27585  0 Aug30 ?        00:00:00 /opt/IBMHTTPServer/bin/httpd
www      27587 27585  0 Aug30 ?        00:00:00 /opt/IBMHTTPServer/bin/httpd
www      27588 27585  0 Aug30 ?        00:00:00 /opt/IBMHTTPServer/bin/httpd
www      27589 27585  0 Aug30 ?        00:00:00 /opt/IBMHTTPServer/bin/httpd
www      27590 27585  0 Aug30 ?        00:00:00 /opt/IBMHTTPServer/bin/httpd
www      28887 27585  0 Aug31 ?        00:00:00 /opt/IBMHTTPServer/bin/httpd
www      28888 27585  0 Aug31 ?        00:00:00 /opt/IBMHTTPServer/bin/httpd
www      28889 27585  0 Aug31 ?        00:00:00 /opt/IBMHTTPServer/bin/httpd
root     13372 13341  0 12:58 pts/5    00:00:00 grep httpd
```

Figure 3-1 Web server sample process listing

2. Check that the IBM HTTP Server is registered to listen on port 80 and is therefore ready to handle requests:

   ```
   netstat -an | grep LISTEN | grep :80
   ```

   The output should look similar to that in Figure 3-2.

```
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN
```

Figure 3-2 Webserver sample socket details

3. If you configured the Web server for SSL, check that the IBM HTTP Server is registered to listen on port 443 and is therefore ready to handle requests:

   ```
   netstat -an | grep LISTEN | grep :443
   ```

   The output should look similar to that in Figure 3-3.

```
tcp        0      0 0.0.0.0:443 0.0.0.0:*               LISTEN
```

Figure 3-3 Web server sample socket details

Check request handling

Using a Web browser, request the following URL representing the IBM HTTP Server Web root for the home page:

```
http://<http_server_hostname>/
```

You should see the IBM HTTP Server home page displayed, as seen in Figure 3-4.

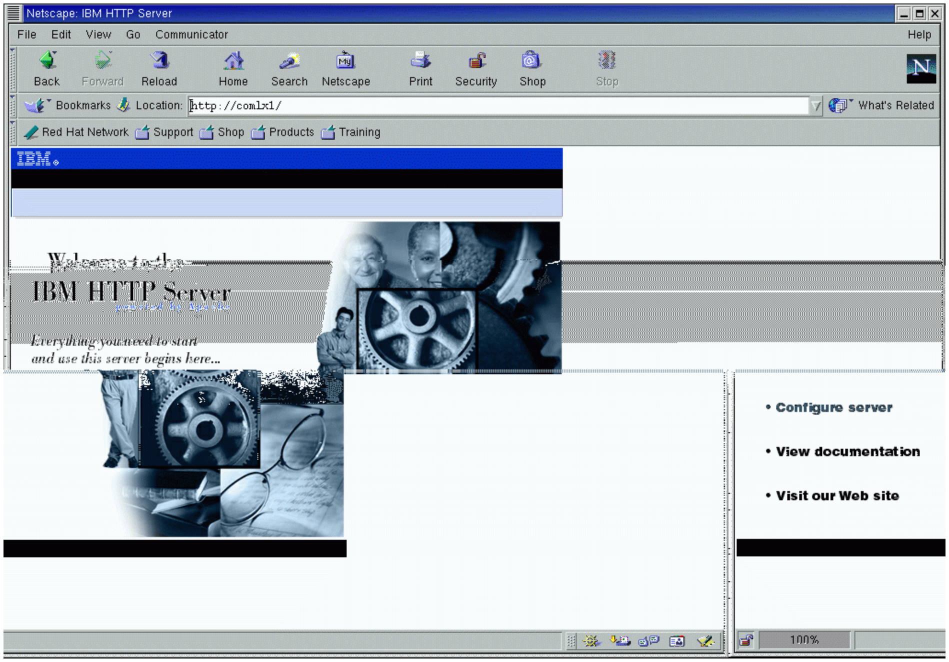

Netscape: IBM HTTP Server
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Shop Stop
Bookmarks Location: http://comlx1/
What's Related
Red Hat Network Support Shop Products Training
IBM.
Welcome to the
IBM HTTP Server
powered by Apache
Everything you need to start
and use this server begins here...
• Configure server
• View documentation
• Visit our Web site
100%

Figure 3-4 IBM HTTP Server home page verification

If your Web server is configured for SSL, repeat the previous step using the following URL:

```
https://<http_server_hostname>/
```

3.5 WebSphere Application Server installation

This section provides detailed instructions for installing, configuring and verifying WebSphere Application Server V4.0.1, Advanced Edition for Linux plus WebSphere Application Server V4 FixPak 4.

The section is organized into the following tasks:

- Pre-installation for WebSphere Application Server
- Install the WebSphere Application Server
- Install WebSphere Application Server V4 FixPak 4
- Install WebSphere Application Server V4.0.4 e-Fixes
- Verify the WebSphere Application Server installation
- Configure the WebSphere Application Server
- Verify the WebSphere Application Server

3.5.1 Pre-installation for WebSphere Application Server

Prior to installing the WebSphere Application Server, the following checks and tasks need to be completed:

- Check that IP ports are unused
- IBM HTTP Server check

Check that IP ports are unused

Check that there are no existing active services that use the following IP ports on the server:

- 900 (bootstrap port)
- 9000 (Naming Service)
- 9080 (default application server)

We suggest using the following command for this task:

```
# netstat -an | grep LISTEN
```

IBM HTTP Server check

The WebSphere Application Server installation utility is capable of installing the IBM HTTP Server during the WebSphere Application Server installation. As an alternative, many system administrators install the IBM HTTP Server prior to the installation of the WebSphere Application Server (as we recommend).

This is done for the following reasons:

- SSL configuration

  If you plan to enable the IBM HTTP Server for SSL, the entries for SSL configuration are included (commented out) in the httpd.conf.sample file and not in the httpd.conf file. The WebSphere Application Server installation utility by default updates the WebSphere Application Server configuration entries to the httpd.conf. The end result is that the user will need to merge entries for WebSphere Application Server and SSL. By installing and configuring the IBM HTTP Server, including SSL configuration prior to the WebSphere Application Server installation, the WebSphere Application Server installation will update the correct version of the httpd.conf with the WebSphere plug-in entries.

- Verify before adding prerequisite components.

  Many administrators strongly believe that each component should be installed, configured, and verified before installing components that rely on the successful installation of the previous component. In this scenario, the user would install the HTTP Server, configure the Web server (SSL, ServerName, etc.), verify the server, and then install WebSphere Application Server.

3.5.2 Install the WebSphere Application Server

There are two ways to install the WebSphere Application Server V4.0.1, Advanced Edition for Linux:

- Interactive install - GUI
- Automated script-based install - silent

To install the IBM WebSphere Application Server V4.0.1, Advanced Edition for Linux using the GUI installer interface, complete the following steps on the WebSphere Application Server server machine:

1. Log in as root and start a terminal session.
2. Load the IBM WebSphere Application Server V4.0 CD-ROM into the CD-ROM drive and mount the CD.

   ```
   # mount -r /dev/cdrom /mnt/cdrom
   ```

3. Change directory to the installation root.
4. Ensure the DISPLAY and TERM environment variables are properly set.
5. Run the install.sh installation script:

   ```
   # ./install.sh
   ```

6. In the Welcome window, click **Next**.

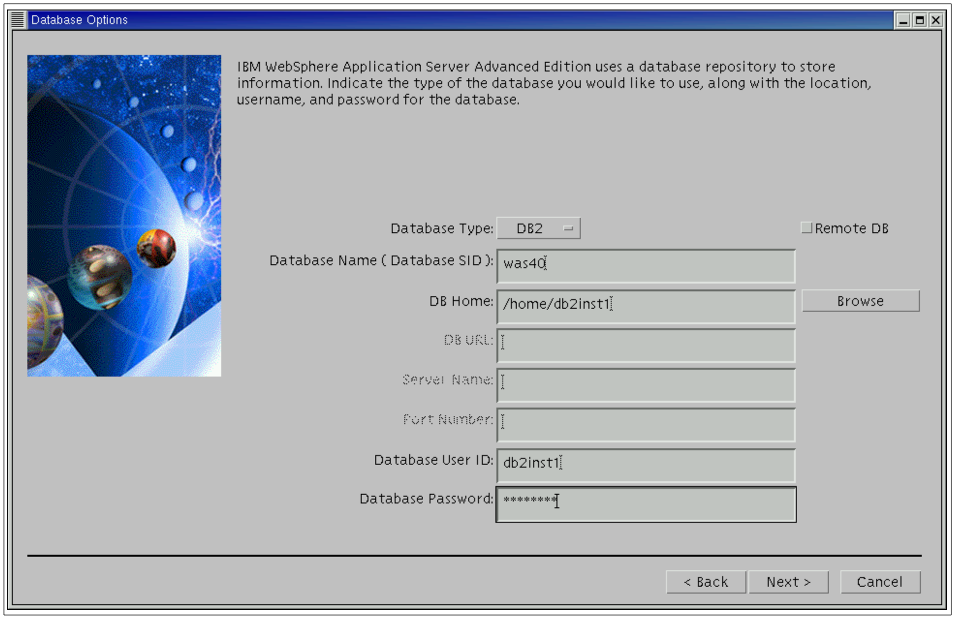
> **Important:** If prerequisite checking is enabled for UNIX, then an alert may be displayed indicating that some of the installation prerequisites have not been met. This can even occur if newer patches/packages than those listed in prereq.properties are installed. If such an alert is displayed, recheck all prerequisites, and if they are met or exceeded, perform the following steps:
>
> - Exit the installation.
> - Disable prerequisite checking.
> - Restart the installation from the beginning.

7. In the Installation Options window, select **Custom Installation** and click **Next**.
8. In the Choose Application Server Components window, select the following and then click **Next**:
   - Server
   - Admin
   - Samples
   - Application Assembly and Deployment tools

In our scenario, we installed the IBM HTTP Server and the Web server plugins first in 3.4, “IBM HTTP Server and WebSphere plugin installation” on page 57 and then installed the WebSphere Application Server separately.

> **Note:** Although not listed in the Application Server Components window, the IBM JDK 1.3.0 is automatically installed under the WebSphere Application Server installation directory. There is no need to separately install a JDK for use by:
>
> - WebSphere Application Server
> - Web Server plug-ins

9. In the Database Options window, enter the following (see Figure 3-5) then click **Next**:



Database Options

IBM WebSphere Application Server Advanced Edition uses a database repository to store information. Indicate the type of the database you would like to use, along with the location, username, and password for the database.

Database Type: DB2 □ Remote DB

Database Name (Database SID): was40

DB Home: /home/db2inst1

DB URL:

Server Name:

Port Number:

Database User ID: db2inst1

Database Password: ********

Figure 3-5 Specify DB2 database connection settings

   – Database Type: `DB2`

   – Database Name (Database SID): `<was_database_alias>`

     Enter the DB2 TCP/IP alias of the WebSphere Application Server repository database created during the configuration of DB2 Server.

   – DB Home: `/home/<db2_instance_owner>`

     For example, /home/db2inst1

- Database User ID: `<db2_instance_owner>`

  For example, db2inst1
- Database Password: `<db2_instance_owner_password>`
- Do *not* select Remote DB.

> **Note:** Whether the DB2 WebSphere Application Server database is local or remote, in our test environment we configure the DB2 client to access the database through a catalog alias. Under these conditions, the Remote DB setting should not be selected.

10. In the Select Destination Directory window, accept the default destination directoy as follows, and then click **Next** to continue:
    - WebSphere Application Server: `/opt/WebSphere/AppServer`
11. In the Install Options Selected window, review the selected components. If they are correct, click **Install** to start the installation.
12. In the Location of Configuration Files window, enter the path to the IBM HTTP Server configuration file (httpd.conf), then click **Next**.
13. In the Setup Complete window, click **Finish**.

3.5.3 Install WebSphere Application Server V4 FixPak 4

This section describes the high-level steps required to install WebSphere Application Server V4 FixPak 4, Advanced Edition.

1. Before you begin the FixPak 4 installation, ensure that the following prerequisites have been met:
   - WebSphere Application Server V4 FixPak 4, Advanced Edition can be installed on top of WebSphere Application Server V4.0.1 or V4.0.2 or V4.0.3, Advanced Edition.
   - Ensure that no WebSphere Application Server processes are running. There should not be any if you have just finished installing WebSphere Application Server (this only applies to the Commerce Application Server node).

     ```
     # ps -ef | grep java
     ```

     The WebSphere Administrative Server can be stopped by selecting the node and right-clicking **stop** from the WebSphere Administrative Console. Alternatively, use the **`wscp`** command line interface to stop the server:

     ```
     # /opt/WebSphere/AppServer/bin/wscp.sh
     wscp > Node stop /Node:comlx1/
     wcsp > exit
     ```

- Ensure that the IBM HTTP Server and Administration Server are stopped as follows:

  ```
  /opt/IBMHTTPServer/bin/apachectl stop
  /opt/IBMHTTPServer/bin/adminctl stop
  ```

- The FixPak 4 Readme states that 110 MB of disk space must be available in the /tmp file system.

> **Note:** We found that if you copy the FixPak 4 tar file and want to uncompress it to the /tmp directory, you will need approximately 150 Mb of free space.

2. Download WebSphere Application Server V4 FixPak 4, Advanced Edition for Linux from to the /tmp directory:

   ```
   http://www-1.ibm.com/support/docview.wss?rs=180&context=SSEQTP&q=FixPak%2B4
   &uid=swg24001635
   ```

   > **Note:** Ensure that you download the Advanced Edition, and not the Single Server Edition.

   > **Note:** For detailed installation instructions, refer to the FixPak 3 Readme file.

3. Decompress the tar file.

   ```
   # cd /tmp
   # tar -xvf was40_ae_ptf_4_linux.intel.tar
   ```

4. To start the FixPak 4 installer, enter the following from the directory the FixPak 4 has been decompressed:

   ```
   # cd /tmp
   # ./install.sh
   ```

5. Accept the licencse agreement by typing `Accept` and pressing **Enter**.
6. We selected the following options:
   - Update Application Server
   - Update JDK
   - Update IBM HTTP Server (not necessary for remote Web server node)
   - We did install the Connector Components (only necessary if you plan to use the connector components)
   - Use Application Server logs directory
   - Place backup in the WebSphere Application Server directory

7. When the FixPak 4 installation is complete, review the logs found in the /opt/WebSphere/AppServer/logs directory:

   – Summary_ptf_4_<date_time_stamp>.log
   – ihs_ptf_4_log
   – install.log
   – jdk_ptf_4.log
   – was40_ae_ptf_4.log

   > **Note:** After we installed the FixPak 4, we discovered that SSL was not longer working on the IBM HTTP Server. To address this issue, continue to the next step.

3.5.4 Install WebSphere Application Server V4.0.4 e-Fixes

This section lists e-Fixes to be installed on top of WebSphere Application Server V4.0.4. Refer to the IBM WebSphere Application Server support page for details:

```
http://www-1.ibm.com/support/search.wss?rs=180&tc=SSEQTP&dc=D400
```

- WebSphere plugin e-Fix (PQ63508)

  Refer to 3.4.4, “Cumulative WebSphere plugin e-fix (PQ63508)” on page 61.

- WebSphere Security e-Fix
- WebSphere Dynamic Caching e-Fix (PQ66335)

After WebSphere Application Server FixPak V4.0.4 is installed, do the following:

1. Open a Web browser and enter the following URL:

   ```
   http://www.ibm.com/software/webservers/appserv/support.html
   ```

2. Select **Support Downloads** -> **All efixes, fixpaks and tools**.
3. Search for and download the following e-Fixes: PQ63116 for AE, PQ61462 and WebSphere plugin cumulative for Linux.
4. Install them as instructed by their readme files. For example, we did the following:

   a. For PQ63116:

      Log in as root, change to the directory of the jar file, and enter:

      ```
      # java -jar
      WAS_Security_10-07-2002_4.0.4-4.0.3-4.0.2_AE_cumulative_eFix.jar
      ```

      When prompted, enter the following:

      ```
      /opt/WebSphere/AppServer
      ```

b. For PQ61462:

   Log in as root, change to the directory of the jar file, and enter:

   ```
   # java -jar PQ61462_eFix.jar
   ```

   When prompted, enter the following:

   ```
   /opt/WebSphere/AppServer
   ```

c. For WebSphere plugin cumulative:

   Log in as root, change to the directory of the jar file, and enter:

   ```
   # java -jar
   WAS_Plugin_10-18-2002_4.0.x_cumulative_eFix_AEServer_AEsServer.jar
   ```

   When prompted, enter the following:

   ```
   /opt/WebSphere/AppServer
   ```

3.5.5 Verify the WebSphere Application Server installation

In order to verify the installation of IBM WebSphere Application Server V4.0.1, Advanced Edition for Linux, the following tasks should be completed:

- Check installation log
- Check admin.config settings:
  `file name /opt/WebSphere/AppServer/bin/admin.config`
- Check Web server configuration file changes

Check the installation log

Check that the installation log /opt/WebSphere/AppServer/logs/install.log does not contain any errors.

Check the admin.config settings

1. Check that the repository database settings are correct for the database type (DB2), instance (was) and user ID (<db2_instance_owner>) used in our test environment:

   ```
   com.ibm.ejs.sm.adminServer.dbdataSourceClassName=COM.ibm.db2.jdbc.DB2Connec
   tionPoolDataSource
   com.ibm.ejs.sm.adminServer.dbserverName=
   com.ibm.ejs.sm.adminServer.dbportNumber=
   com.ibm.ejs.sm.adminServer.dbdatabaseName=<was database alias>
   com.ibm.ejs.sm.adminServer.dbuser=<db2_instance_owner>
   com.ibm.ejs.sm.adminServer.dbpassword=<db2owner_password>
   com.ibm.ejs.sm.adminServer.dbdisable2Phase=true
   ```

2. Check that the following path-related parameters are set correctly:

Table 3-10 DB2-related paths required in admin.config

Parameter	Must contain path...
com.ibm.ejs.sm.util.process.Nanny.adminServerJvmArgs	<db2_install_path>/sqllib/java12/db2java.zip

3. Check that the WebSphere Application Server schema and initial configuration (for example, Default Server) will be written to the repository database on startup:

Table 3-11 Required schema creation and initial configuration flags

Parameter	Required value...
com.ibm.ejs.sm.adminServer.createTables	true
install.initial.config	true

Check Web server configuration file changes

Verify that the Web server configuration file has been updated by completing the following steps:

1. Check that the following required settings have been added to the IBM HTTP Server configuration file (httpd.conf) as a result of the WebSphere Application Server installation:

```
LoadModule ibm_app_server_http_module
/opt/WebSphere/AppServer/bin/mod_ibm_app_server_http.so
WebSpherePluginConfig /opt/WebSphere/AppServer/config/plugin-cfg.xml
AddModule mod_app_server_http.c
```

2. If not, manually add the above lines to the end of the httpd.conf file and save the changes.

3.5.6 Configure the WebSphere Application Server

This section describes how to configure the WebSphere Application Server and includes the following tasks:

- Start up IBM HTTP Server processes
- Start up WebSphere Admin Server processes
- Configure WebSphere Application Server hosts aliaes

Start up IBM HTTP Server processes

The IBM HTTP Server processes need to be started in order to serve content. For details, refer to 3.4.6, “Restart, stop and start IBM HTTP Server” on page 67.

Start up WebSphere Admin Server processes

The WebSphere Application Server Admin Server needs to be started in order to test the installation as well as the connectivity between WebSphere Application Server Admin Server, WebSphere Application Server repository database, and the Administrator Console.

1. Log in as root and start a terminal session.
2. Ensure that DB2 is started for the DB2 instance containing the WebSphere Application Server repository database. For example:

   ```
   # su - db2inst1
   $ db2start
   $ exit
   ```

3. To start the WebSphere Admin Server, type the following commands:

   ```
   # cd /opt/WebSphere/AppServer/bin
   # ./startupServer.sh
   ```

4. Verify that the startup of WebSphere Application Server Admin Server is successful by reviewing the the following conditions:

   a. There are no Admin Server error logs in <was_home>/logs with names that start with `__adminServer`.

   b. The last line of the /opt/WebSphere/AppServer/logs/tracefile file is similar to the following:

   ```
   [01.07.05 11:28:01:591 EDT] 160042d2 Server I WSVR0023I: Server
   __adminServer open for e-business
   ```

5. Check that the WebSphere Admin Server process is running by issuing the following command:

   ```
   # ps -ef | grep java
   ```

   The output should list a number of Java processes.

> **Note:** In the event that more than one Java process is running unintentionally, you can stop the process by invoking the **`kill`** command:
>
> ```
> kill <process_id>
> ```

Configure WebSphere Application Server hosts aliaes

Many runtime environments include the Web server being configured to support SSL connections. In this case, we need to configure host aliases for the virtual hosts for the WebSphere Application Server.

1. Start the WebSphere Administrator Console as follows:

   ```
   # cd /opt/WebSphere/AppServer/bin
   # ./adminclient.sh
   ```

2. Select the **Virtual Hosts** folder.
3. Add the virtual host aliases as listed in Table 3-12 if they are not already added. Click **Add** to add them, and then click **Apply**.

Table 3-12 WebSphere Application Server host aliases for virtual host

Virtual host	Example
<hostname>:80	comlx1:80
<fully_qualified_hostname>:80	comlx1.itso.ral.ibm.com:80
<ip_address>:80	9.24.105.112:80
<ip_address>:443	9.24.105.112:443
<hostname>:443	comlx1:443
<fully_qualified_hostname>:443	comlx1.itso.ral.ibm.com:443

3.5.7 Verify the WebSphere Application Server

This section describes how to verify the functionality of the WebSphere Application Server and includes the following tasks:

- Start WebSphere Application Server Default Server
- Regenerate Web server plug-in settings
- Restart Web server processes
- Verify Web server plug-in and WebSphere configuration

Start WebSphere Application Server Default Server

The WebSphere Application Server installation sets up a default application server (Default Server) in the WebSphere Application Server administrative domain. This Application Server and its servlets are used to verify that the WebSphere Application Server installation is working correctly.

To start the Default Server, perform the following steps:

1. Start the WebSphere Administrative Console by issuing the following commands:

```
cd /opt/WebSphere/AppServer/bin
./adminclient.sh
```

2. Select and expand **WebSphere Administrative Domain -> Nodes -> <your_hostname> -> Application Servers**.
3. Select **Default Host**, and then right-click **Start**, if not already started.
4. The startup of Default Server is successful if the following condition is met:

– The AdminConsole event messages pane shows the lines:

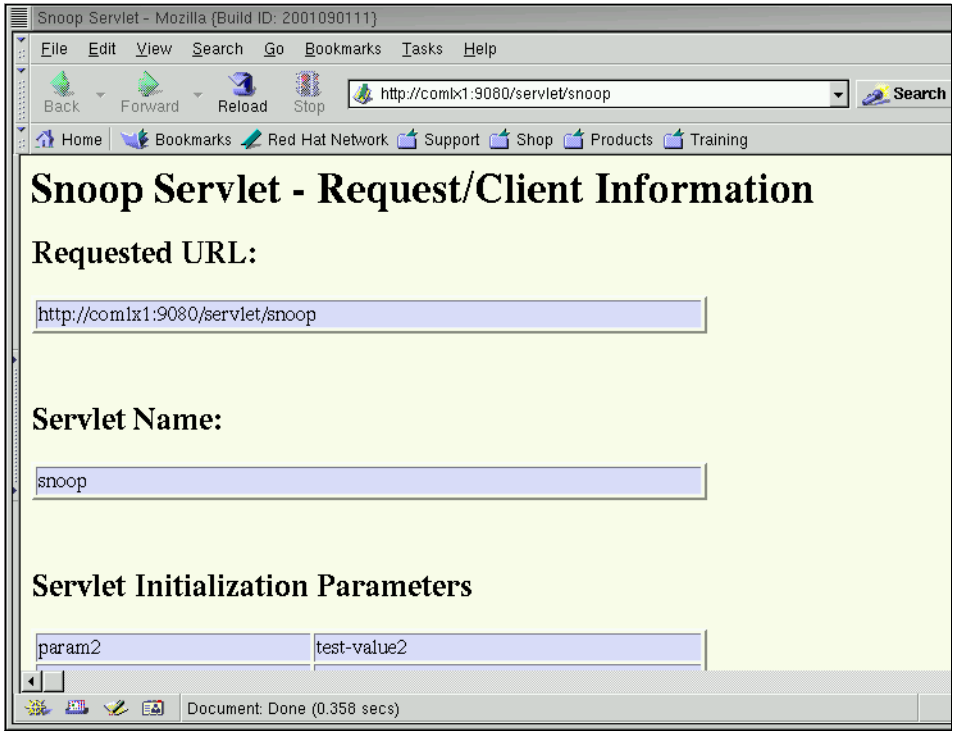
```
Command “Default Server.start” completed successfully.
Transport http listening on port 9080.
```

5. Verify that the Default Server Web container has been properly installed and configured by accessing its servlets through the Web server “embedded” within the WebSphere Application Server V4 Web container:

   a. Using a Web browser, request the following URL:

   ```
   http://<hostname>:9080/servlet/snoop
   ```

   A page similar to the one shown in Figure 3-6 should be displayed.



Snoop Servlet - Request/Client Information

Requested URL:

http://comlx1:9080/servlet/snoop

Servlet Name:

snoop

Servlet Initialization Parameters

param2	test-value2

Document: Done (0.358 secs)

Figure 3-6 Snoop servlet accessed through embedded Web server

   b. Using a Web browser, request the following URL:

   ```
   http://<was_server_hostname>:9080/webapp/examples/showCfg
   ```

Note: The embedded Web server is a new feature introduced with WebSphere Application Server V4.0. In previous releases, a stand-alone Web server was required in order to access any resource hosted in WebSphere Application Server.

Regenerate Web server plug-in settings

Before the Default Server can be accessed from a stand-alone Web server (for example, the IBM HTTP Server) the Web server plug-in settings file <was_home>/config/plugin-cfg.xml must be regenerated to reflect the following settings used by the Web server plug-in:

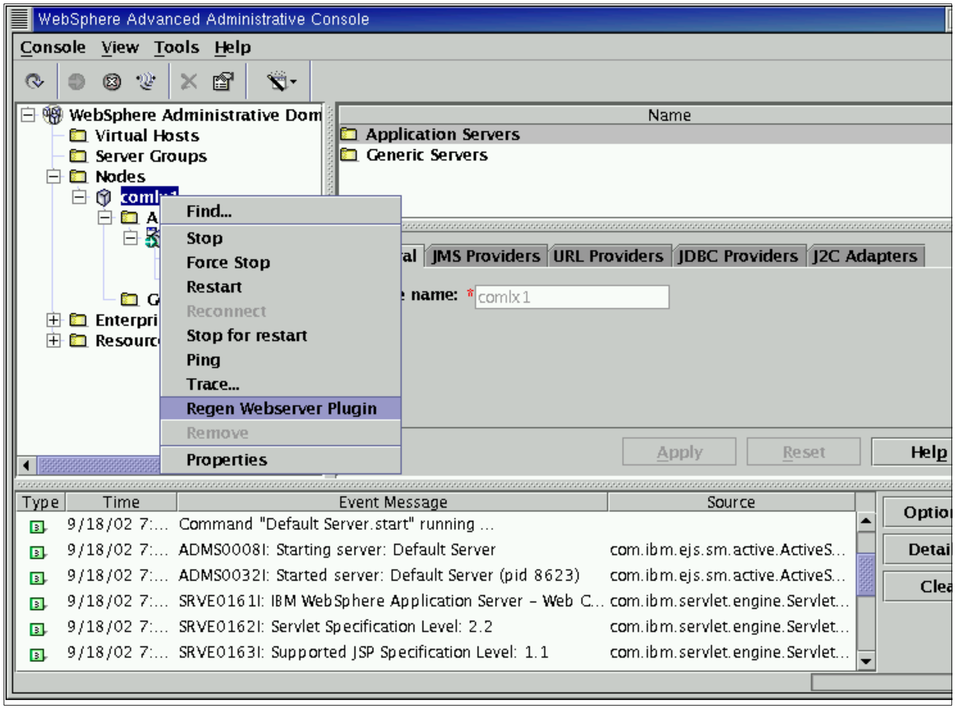
- Virtual host settings
- Application Server transports
- Web Container URIs

Perform the following steps:

1. If you exited the WebSphere Administration Console:
   a. Log in as root, and start a terminal session.
   b. Run the WebSphere Application Server Administrative Console by issuing the following command:

```
# cd /opt/WebSphere/AppServer/bin
# ./adminclient.sh &
```

2. Select and expand **WebSphere Administrative Domain -> Nodes**.
3. Select **<your_hostname>**, right-click and select **Regen Webserver Plugin** (see Figure 3-7).



name: * comlx1

Type	Time	Event Message	Source
	9/18/02 7:...	Command "Default Server.start" running ...	
	9/18/02 7:...	ADMS0008I: Starting server: Default Server	com.ibm.ejs.sm.active.ActiveS...
	9/18/02 7:...	ADMS0032I: Started server: Default Server (pid 8623)	com.ibm.ejs.sm.active.ActiveS...
	9/18/02 7:...	SRVE0161I: IBM WebSphere Application Server - Web C...	com.ibm.servlet.engine.Servlet...
	9/18/02 7:...	SRVE0162I: Servlet Specification Level: 2.2	com.ibm.servlet.engine.Servlet...
	9/18/02 7:...	SRVE0163I: Supported JSP Specification Level: 1.1	com.ibm.servlet.engine.Servlet...

Figure 3-7 Regenerate Web server plug-in settings

Tip: WebSphere Application Server provides a command-line tool that can be used to regenerate the Web server plug-in configuration without having to run the GUI AdminConsole:

```
/opt/WebSphere/AppServer/bin/GenPluginCfg.sh -adminNodeName <hostname>
```

4. Check that the content of the <was_home>/config/plugin-cfg.xml file has been updated to include the URIs of servlets contained within Default Server.

Tip: The plug-in regeneration command generates a <Server> element for Default Server that contains a CloneID attribute:

```
<Server CloneID="u1uk9fut" Name="Default Server">
   <Transport Hostname="comlx1" Port="9080" Protocol="http"/>
</Server>
```

In a non-cloned environment, this attribute can be removed, resulting in performance improvements by the Web server plug-in.

Restart Web server processes

The IBM HTTP Server process must be restarted before the Web server plug-in configuration can be tested after being regenerated (actually on time refresh). Refer to 3.4.6, “Restart, stop and start IBM HTTP Server” on page 67 for details.

Verify Web server plug-in and WebSphere configuration

The Web server plug-in configuration can be verified by requesting a servlet through the Web server that has already been successfully requested through the Web container’s embedded Web server. Using a Web browser, request the following URL:

```
http://<web_server_hostname>/webapp/examples/showCfg
```

3.6 WebSphere Commerce installation

This section describes the high-level steps to install and configure WebSphere Commerce V5.4. For more detailed information, refer to the *Installation Guide, IBM WebSphere Commerce V5.4 Professional and Business Edition for Linux* product guide.

This section describes the high-level steps to install WebSphere Commerce V5.4 for Linux.

1. Make sure the following are started:
   - IBM DB2 (db2start)
   - IBM WebSphere Administrative Server (startupServer.sh)
2. Mount the WebSphere Commerce V5.4 CD.

   ```
   # mount -r /dev/cdrom /mnt/cdrom
   # cd /mnt/cdrom
   ```

3. On the command line, type:

   ```
   # ./install.sh
   ```

4. After entering install.sh you will be prompted with a message stating that `WebSphere Application Server V4.0.2 is required`. In our example, we have already installed a higher FixPak level, WebSphere Application Server V4 FixPak 4 (or 3). Type `Y` and press **Enter** to continue.
5. When the Welcome window appears, click **Next**.
6. When the License Agreement window appears, review the agreement, select **I accept the terms in the license agreement**, and click **Next**.
7. You will see a message: `the IBM WebSphere Commerce V5.4, Business Edition will installed in the /opt/WebSphere/CommerceServer`

`directory for a total size of 1693.9 MB`. Click **Next** to begin the installation.

8. When the installation is complete, click **Finish**.
9. After clicking **Finish**, you will be prompted with the following message:

   ```
   If WPM (Server and Cassettes) is to be installed on same machine as WC,
   please press y :
   ```

   For our single-tier scenario, we answered `y`. This will cause the installer to exit and explain how to continue to the manual WebSphere Payment Manager install (renamed to WebSphere Commerce Payments). Continue to 3.7, “WebSphere Commerce Payments installation” on page 84.

   > **Note:** The WebSphere Commerce installer on Linux is designed to run the wcb.sh script if you answer `n` (install WebSphere Payment Manager on a separate node). The ordering of installation components requires this prior to the wcb.sh script being run.
   >
   > We prefer to run the wcb.sh script manually when needed to incorporate higher FixPak levels beyond what is shipped with the product.

3.7 WebSphere Commerce Payments installation

After IBM WebSphere Commerce V5.4, Business Edition was released, the IBM WebSphere Payment Manager V3.1.2 product included with WebSphere Commerce, was renamed to WebSphere Commerce Payments V3.1.3 and made available for download as a FixPak upgrade.

This section describes how to install IBM WebSphere Commerce Payments V3.1.3 within a WebSphere Commerce single-tier runtime environment. This is a two-stage process, which begins with the installation of IBM WebSphere Payment Manager V3.1.2 from the CD included in the WebSphere Commerce V5.4 package. Next, we will download and install the WebSphere Commerce V3.1.3 FixPak. The upgraded level with be WebSphere Commerce Payments V3.1.3.

Note: For additional information on installing and configuring WebSphere Payment Manager, see the following:

- *Installation Guide, IBM WebSphere Payment Manager for Multiplatforms* on the Payment Manager CD.
- *Administrator's Guide, IBM WebSphere Payment Manager for Multiplatforms* on the Payment Manager CD.

Once you have completed your install, refer to the WebSphere Commerce V5.4 online documentation which contains information about configuration with your WebSphere Commerce store.

3.7.1 Pre-installation for WebSphere Payment Manager V3.1.2

Prior to installing IBM WebSphere Payment Manager V3.1.2, ensure the following prerequisites steps have been completed.

1. Create the WebSphere Payment Manager database on the database server from the DB2 command line using the following commands:

   Syntax:

   ```
   > db2 create db <payment_db_name>
   ```

   Example:

   ```
   # su - db2inst1
   $ db2 create db payman
   ```

2. Update the WebSphere Payment Manager database applheapsz:

   Syntax:

   ```
   > db2 update db cfg for <payment_db_name> using applheapsz 256
   ```

   Example:

   ```
   $ db2 update db cfg for payman using applheapsz 256
   ```

3. List the DB2 databases catalogued for this node.

   ```
   $ db2 list db directory
   ```

4. Catalog the WebSphere Payment Manager database on the WebSphere Commerce Server system using the following DB2 command:

   Syntax:

   ```
   > db2 catalog db <payment_db_name> as <payment_db_alias> at node
   <db_server_node_name>
   ```

   Example:

   ```
   $ db2 catalog db payman as payman1 at node comlx1
   ```

   Where `comlx1` is the hostname of the DB2 Server.

5. Catalog the TCP/IP node. The db2inst1 should have already had the **`tcpip catalog`** command executed when creating the WebSphere repository.

   In order to access the administration database via TCP/IP, the DB2 node must first be catalogued. In our scenario, we have already catalogued the tcpip node in 3.3.6, “Create the WebSphere Application Server repository database” on page 54.

   For example:

   ```
   $ db2 catalog tcpip node comlx1 remote comlx1 server db2cdb2inst1
   ```

   Syntax:

   ```
   db2 catalog tcpip node <node_name> remote <local_hostname> server
   <service_name>
   ```

   The `<service_name>` used to catalog the node must be the same as the database instance connection port name in the /etc/services file. The `<node_name>` chosen can be any valid DB2 node name.

6. Make sure the following are started:

   – IBM DB2 (db2start)

   – IBM WebSphere Administrative Server (startupServer.sh) as a root user (port 900) during the WebSphere Payment Manager installation.

7. Check that the /etc/hosts file contains the loopback and your server host IP address entries. For example:

   ```
   127.0.0.1 localhost.localhostdomain localhost
   9.24.105.112 comlx1.itso.ral.ibm.com comlx1
   ```

Note: Payment Manager Installer will not proceed if the network setting is not correct.

3.7.2 Install WebSphere Payment Manager V3.1.2

To install WebSphere Payment Manager, complete the following steps:

1. Start a Linux Console and log in as root.
2. Insert the IBM WebSphere Payment Manager V3.1.2 CD in the CD-ROM drive and mount the CD.

   ```
   # mount -r /dev/cdrom /mnt/cdrom
   # cd /mnt/cdrom
   ```

3. To start the install, enter the following from the IBM WebSphere Payment Manager V3.1.2 CD.

   ```
   # cd linux
   # /opt/WebSphere/AppServer/java/bin/java -classpath install.jar:xml4j.jar
   Installer
   ```

   > **Note:** We found that the Install program supplied with IBM WebSphere Payment Manager V3.1.2 did not work properly within our runtime environment. For this reason, we started the Java-based WebSphere Payment Manager installer manually using the JRE supplied with the WebSphere Application Server.

4. When the Welcome window appears, click **Next**.
5. When the Choose Destination Directory window appears, accept the default location (**/opt/PaymentManager**) and click **Next**.
6. When the Database Product Selection window appears, select **IBM Universal Database** and then click **Next**.
7. When the JDBC Driver Information window appears, enter the following and then click **Next**:
   - JDBC Driver Class Name: `COM.ibm.db2.jdbc.app.DB2Driver` (default)
   - Location of JDBC Classes: `/usr/IBMdb2/V7.1/java12/db2java.zip` (default)
   - JDBC Driver Shared Library Path: `/usr/IBMdb2/V7.1/java12/`
   - DB2 Instance Name: `db2inst1`
8. When the Payment Manager Database Access Information window appears, enter the following and then click **Next**:
   - Database Owner User ID: `db2inst1`
   - Database Administrator’s User ID: `db2inst1`
   - Database Administrator’s Password: `<your_password>`
   - Database Manager Database Name: `payman1`

Where `payman1` is the database alias name created in 3.7.1, “Pre-installation for WebSphere Payment Manager V3.1.2” on page 85

9. When the Payment Manager WebSphere Configuration Information window appears, enter the following and then click **Next**:
   – WebSphere Administrative Domain Name: `comlx1`
10. When the Installation Summary window appears, review the information and then click **Next** to begin copying files.

    The installation takes approximately five minutes.
11. When the IBM WebSphere Payment Manager, Version 3.1.2.0 Readme window appears, click **Next**.
12. Unmount the CD-ROM drive and remove the IBM Payment Manager 3.1.2 CD.

3.7.3 Install WebSphere Commerce Payments V3.1.3 FixPak

As mentioned, service updates for WebSphere Commerce Payments can be downloaded from the IBM Web site (this requires user registration) at:

```
https://www6.software.ibm.com/dl/paymgr/srvupdts-p
```

Updates for the various cassettes are also available. The download file containing the update for the WebSphere Commerce Payments Framework is called IBMPayMgr_linux_Framework_Ctr_3_1_3_0_ptf.class (34 MB).

1. Start a Linux Console and log in as root.
2. Download the IBMPayMgr_linux_Framework_Ctr_3_1_3_0_ptf.class file to a temporary directory, to which you have write access, such as /tmp.
3. Ensure IBM WebSphere Application Server is running under the root user, and on the default bootstrap port of 900 (there should not be a com.ibm.ejs.sm.adminServer.bootstrapPort parameter specified in the /opt/WebSphere/AppServer/bin/admin.config file).
4. Stop the WebSphere Payment Manager application server using the WebSphere Application Server Administrative Console.
5. Make sure the directory containing the ptf.class file is in the CLASSPATH environment variable (for example we dowloaded the FixPak pft.class to the /tmp directory):

   ```
   CLASSPATH=/tmp:$CLASSPATH
   export CLASSPATH
   ```

6. To start the PTF installer, enter this command (text below wraps) from the directory where you downloaded it:

```
# /opt/WebSphere/AppServer/java/bin/java
IBMPayMgr_linux_Framework_Ctr_3_1_3_0_ptf
```

Note:

- Do not include .class in the file name when starting the PTF installer.
- As an alternative to starting Java by entering /opt/WebSphere/AppServer/java/jre/bin/java, you can set the JAVA_HOME and PATH as follows and then run Java:

```
# export JAVA_HOME=/opt/WebSphere/AppServer/java/bin
# export PATH=$PATH:$JAVA_HOME
# java <ptf.class>
```

7. At the WebSphere Commerce Payments 3.1.3.0 PTF window, click **Next**.
8. When prompted to enter the password to the database payman1, we entered the db2inst1 (DB2 instance owner) <password> and then clicked **Next**. If WebSphere Application Server is not using the default bootstrap port, the PTF installer will claim that it is not running.

   The installation takes approximately five minutes.
9. When the Upgrade Complete window appears, click **Next**.

The WebSphere Commerce Payments installation is now complete.

3.8 WebSphere Commerce post-install configuration

This section describes procedures that need to be completed after the WebSphere Commerce installation and before instance creation.

The section is organized into the following tasks:

- Execute post install script (wcb.sh)
- Post wcb.sh configuration
- Start WebSphere Administrative Server (non-root wasuser)
- Start WebSphere Administrative Console (non-root wasuser)
- WebSphere Commerce FixPak V5.4.0.3 installation
- Verify WebSphere Application Server non-root user
- Modify was.deployed.EJB.xml (workaround)
- Update setenv.sh (workaround)
- Change file ownership and permissions (workaround)

3.8.1 Execute post install script (wcb.sh)

After the WebSphere Commerce code has been installed, it is necessary to set up the environment for WebSphere Commerce by executing the wcb.sh script. This script will also create an operating system user ID that will allow the WebSphere Application Server, Payment Manager and WebSphere Commerce as a non-root user. Ownership of the WebSphere Application Server and WebSphere Commerce directories will also be changed to this new user.

On all other platforms, the post install script is named wcpostinstall.sh. On the Linux platform, this script is named wcb.sh.

> **Note:** Full instructions for running the postinstall script section are provided in the *Installation Guide, IBM WebSphere Commerce V5.4 Professional and Business Edition for Linux* product guide.

In order to run as a non-root user, the WebSphere Application Server bootstrap port (the default is 900) must be set to a non-privileged port number (greater than or equal to 1024). The wcb.sh script chooses port 2222 by default. You can specify any available port, and also change it at a later time, by editing the com.ibm.ejs.sm.adminServer.bootstrapPort setting in the /opt/WebSphere/AppServer/bin/admin.config file.

To execute the wcb.sh script, do the following:

1. Ensure that the following applications processes are stopped:
   - DB2 Server
   - WebSphere Application Server Administrative Server
   - IBM HTTP Server (not necessary for remote Web server scenario)
2. Run the wcb.sh script in the /opt/WebSphere/CommerceServer directory:

```
# cd /opt/WebSphere/CommerceServer/bin
# ./wcb.sh
```

3. Follow the on screen prompts, ensuring that you select **yes** when asked if you want to run as a non-root user. In this example, we used the settings listed in Table 3-13.

Table 3-13 Settings for wcb.sh

Attribute	**Value for single-tier**
Language locale	en_US
Short hostname	comlx1
Domain	itso.ral.ibm.com

Attribute	Value for single-tier
WebSphere Application Server home	/opt/WebSphere/AppServer
Database	**1** DB2 UDB 7.2 FP6 Note: In our example, we have installed DB2 FixPak 7. You will get an error message stating the level detected is unsupported. Disregard this message and continue.
Database install directory	/usr/IBMdb2/V7.1
Database user	db2inst1
Primary group	db2iadm1
Web server	**1** IBM HTTP Server 1.3.19.1 (for local Web server) **4** for remote Web server
IBM HTTP Server path	/opt/IBMHTTPServer
Payment Manager	**1** for local Payment Manager **2** for remote Payment Manager
Payment Manager directory	/opt/PaymentManager
Non-root user	**1** (Yes)
User	wasuser
User group	wasgroup
Admin port	2222
EFix Installation	2 No (we have already installed higher levels of the required FixPaks).

4. After making all selections, you will see a listing of options selected. The text to explain how to proceed is at the top of the listed options (scroll up). Type `1` and press **Enter** to begin the script processing.

   > **Note:** The window in this step can be very deceiving. The question asking you to press **1** to begin processing is at the top of the page. You must scroll up to see the message.

5. At the end of the script, you will be prompted to `set the wasuser` password. You may see an error message: `BAD PASSWORD`. It does not appear to matter

what is entered; you will still get this message. You may need to set this password manually after the script is run using the **passwd** command.

6. Verify the password is set correctly for the wasuser.

   ```
   # su - wasuser
   $ su - wasuser
   Password: <your_password>
   $ exit
   $ exit
   ```

7. Before using the graphical terminal set in the DISPLAY variable, it may be necessary for the root user to add the hostname to the X-Windows access control list (if remote):

   ```
   # /usr/bin/X11/xhost <hostname>
   ```

8. After running the wcb.sh script, you must update the .bashrc file for the new wasuser by adding the following lines:

   ```
   DISPLAY=<hostname.domain>:0.0
   export DISPLAY
   ```

   The updated PATH is required for the WebSphere Commerce Configuration Manager. The DISPLAY variable should be set to the hostname of the graphical terminal which will be used to administer WebSphere Application Server and WebSphere Commerce. Setting a default value in the .profile eliminates the need to manually set it each time a graphical terminal is required.

   > **Note:** Remember to change the DISPLAY setting if you work from a different machine later.

9. Restart the DB2 Server (ensure it is started, and if not, start it).

3.8.2 Post wcb.sh configuration

This section describes several very important updates needed for the non-root user (for example, wasuser) to run the WebSphere Application Server and WebSphere Commerce Sever properly.

When the wcb.sh post install script is executed, it creates a non-root user, updates the adminServer port, changes file ownership and permissions, and much more. Unfortunately, it does not update the non-root user .bashrc to include support for DB2 (sqllib, usejdbc2), and does not set the properly Linux shell. It creates the non-root user with the Korn shell (ksh).

Also, to address a JIT complile error with the WebSphere Commerce Loader, we have added a procedure to exclude the Loader from being JIT compiled.

Update the non-root user (wasuser) shell in passwd

To update the non-root user (wasuser) shell to use the Bash shell instead of the Korn shell, update the /etc/passwd file as follows:

1. Check which shell is being used by the non-root user (wasuser) as follows:

   ```
   # su - wasuser
   $ chsh
   $ exit
   ```

2. To update the default shell for the non-root user (wasuser) created by wcb.sh, edit the /etc/passwd file and change the shell entry to /bin/bash:

   Original:

   ```
   wasuser:x:506:506:NonrootUser:/home/wasuser:/bin/ksh
   ```

   Modified:

   ```
   wasuser:x:506:506:NonrootUser:/home/wasuser:/bin/bash
   ```

Update the non-root user .bashrc for DB2

To update the non-root user (wasuser) .bashrc for DB2, which is needed for the WebSphere Commerce Loader (instance creation, store publishing, etc), do the following:

1. Log in as the non-root user (wasuser):

   ```
   # su - wasuser
   ```

2. Add the following DB2 environment setup and the use of JDBC2 to the /home/wasuser/.bashrc directory.

   ```
   # Setup DB2 environment
   if [ -f /home/db2inst1/sqllib/db2profile ]; then
   . /home/db2inst1/sqllib/db2profile
   fi

   # Setup JDBC2
   if [ -f /home/db2inst1/sqllib/java12/usejdbc2 ]; then
   . /home/db2inst1/sqllib/java12/usejdbc2
   fi
   ```

Add JITC_COMPILEOPT to the /etc/profile

When using WebSphere Application Server V4 FixPak 3 or 4, the WebSphere Commerce Loader does not work properly due to a JIT compiler problem.

Prior to using the Loader for instance creation or publishing a store, it is necessary that you add the JITC_COMPILEOPT environment variable so the MassLoader will work properly. This issue was introduced with the WebSphere Application Server V4 FixPak 3 and still exists in FixPak 4.

To address this WebSphere Commerce Loader issue, enter the following environment variable to the /etc/profile:

```
export
JITC_COMPILEOPT=SKIP{com/ibm/wca/MassLoader/Formatter/JDBCFormatter$FormatW
orker}{run}
```

> **Note:** The export statement above is wrapped, and should be entered on one line in the /etc/profile.

Add JITC_COMPILEOPT to the “startupServer.sh” file

Enter the same environment variable to the <was_home>/bin/startupServer.sh with single quotes:

```
export
JITC_COMPILEOPT='SKIP{com/ibm/wca/MassLoader/Formatter/JDBCFormatter$Format
Worker}{run}'
```

3.8.3 Start WebSphere Administrative Server (non-root wasuser)

To start the WebSphere Application Server Administrative Server with the non-root user created by wcpostinstall.sh named wasuser, do the following:

1. Ensure the WebSphere Application Server Administrative Server is not running as user root.
2. Start the WebSphere Application Server Administrative Server as non-root user wasuser as follows:

   ```
   # su - wasuser
   $ /opt/WebSphere/AppServer/bin/startupServer.sh &
   ```

   Verfiy that the Administrative Server is started:

   ```
   $ tail -f /opt/WebSphere/AppServer/logs/tracefile
   ```

   If successful, you should see the following message at the end of the tracefile:

   ```
   Server _adminServer open for e-business
   ```

3.8.4 Start WebSphere Administrative Console (non-root wasuser)

After the WebSphere Administrative Server has started, launch the WebSphere Application Server Administrator Console, again as the non-root wasuser. Ensure X-windows DISPLAY has been set (.profile update). From now on, it will be necessary to specify the hostname and port number:

1. Log on as root.
2. Set the xhost access:

   ```
   # xhost <was_host>
   ```

3. Log on as the non-root user (wasuser):

   ```
   # su - wasuser
   ```

4. Start the WebSphere Administrative Console. Ensure the DISPLAY is set. In our example, we update the wasuser .profile with export DISPLAY=comlx1.itso.ral.ibm.com:0.0 in a previous step.

   Syntax:

   ```
   /opt/WebSphere/AppServer/bin/adminclient.sh <was_hostname> <port>
   ```

   Example:

   ```
   $ cd /opt/WebSphere/AppServer/bin
   $ ./adminclient.sh comlx1 2222
   ```

3.8.5 WebSphere Commerce FixPak V5.4.0.3 installation

This section describes the high-level steps necessary to install the WebSphere Commerce FixPak V5.4.0.3 for Linux within this environment.

WebSphere Commerce FixPak V5.4.0.3 was made available for download from the WebSphere Commerce support page. To simplify the procedure for clean installations, we have installed the prerequisite components for the WebSphere Commerce FixPak V5.4.0.3, DB2 UDB V7 FixPak 7 and WebSphere Application Server V4 FixPak 4 (this is not a hard requirement, but it includes e-Fixes that are required).

The WebSphere Commerce FixPak V5.4.0.3 includes the following:

- WC5403_Readme.pdf

  This PDF describes what was fixed and how to install the FixPak for Windows NT/2000, AIX, Solaris and Linux.

- WC5403_LINUX.tar

  This zip file contains fixes for WebSphere Commerce V5.4. After installing the FixPak, the level of WebSphere Commerce should be V5.4.0.3.

Note: We recommend that you read the WC5403_Readme.pdf. It contains a listing of the defects fixed as well as detailed installation procedures.

The high-level steps to download and install the FixPak are as follows:

1. Ensure that the following application processes are stopped prior to the FixPak installation if installed on the node where the FixPak will be installed:
   - IBM HTTP Server
   - IBM HTTP Administration
   - IBM WebSphere Administrative Server
   - IBM WC Configuration Manager Server
2. The WebSphere Commerce FixPak V5.4.0.3 for Linux, can be found at the following URL:

   `http://www-1.ibm.com/support/docview.wss?rs=497&uid=swg24001839`
3. Download the following FixPak files to a temporary directory (for example, /tmp/wc5403):
   - WC5403_Readme.pdf
   - WC5403_LINUX.tar
4. Log on as user root and untar the WC5403_LINUX.tar into the same directory.

   ```
   # cd /tmp/wc5403
   # tar -xvf WC5403_LINUX.tar
   ```
5. Start the FixPak installation by typing the following from the command line in the /tmp/`wc5403` directory:

   Syntax:

   ```
   # ./install_wc.sh <WC_HOME> <WAS_HOME> <WC_EDITION>
   ```

   Where `<WC_HOME>` is the WebSphere Commerce install path.

   Where `<WAS_HOME>` is the WebSphere Application Server install path.

   Where `<WC_EDITION>` is the WebSphere Commerce Edition `<Business | Pro>`.

   Note: The edition name is case sensitive. For example, Business must be spelled as is (capital B).

   Example:

   ```
   # ./install_wc.sh /opt/WebSphere/CommerceServer /opt/WebSphere/AppServer
   Business
   ```

Important: If you have already created a WebSphere Commerce instance, for each instance, refer to the WC5403_Readme.pdf for instructions on updating each application server.

6. When the FixPak installation is complete, review the WebSphere Commerce FixPak V5.4.0.3 log files and check for errors/success:

   `<WC_HOME>/service/fixpack/5403/WCfixpack5403_wc_LINUX.log.`

   Also, for each instance, verify the logs for the WebSphere Application Server WebSphere Commerce instance application:

   ```
   <WAS_HOME>\installedApps\WC_Enterprise_App_[instance_name].ear\service\fixp
   ack\5403\WCfixpack<edition>5403_was_LINUX.log.
   ```

   where `<instance_name>` is the deployed WebSphere Commerce instance name. Ensure that no errors are reported. The text “`Completed WebSphere Commerce 5.4.0.3 FixPak Install with no errors`.” should be shown at the end of the log file.

Note:

- Start the WebSphere Administrative Server as non-root user, wasuser. Refer to 3.8.3, “Start WebSphere Administrative Server (non-root wasuser)” on page 94.
- Create a backup of the httpd.conf on the node where the IBM HTTP Server is installed before creating a WebSphere Commerce instance. In the case of a remote Web server, updates to the httpd.conf need to be performed manually.

3.8.6 Verify WebSphere Application Server non-root user

Now that WebSphere Commerce wcpostinstall.sh has modified that Administrative Server to run as a non-root user (wasuser), we recommend that you verify the three-tier functionality of the runtime.

1. Start the following servers:
   - DB2 Server

     Refer to 3.3.8, “Start DB2” on page 56.
   - IBM HTTP Server

     Refer to 3.4.6, “Restart, stop and start IBM HTTP Server” on page 67

- WebSphere Administrative Server

  Refer to 3.8.3, “Start WebSphere Administrative Server (non-root wasuser)” on page 94

2. Start the WebSphere Adminstrative Console as a non-root user, wasuser.

   Refer to 3.8.4, “Start WebSphere Administrative Console (non-root wasuser)” on page 95.

3. Start the WebSphere Payment Manager Server application server from the WebSphere Administrative Console.

   Refer to 3.11.1, “Start the WebSphere Commerce Payments Server” on page 110.

4. Start the Default Server application server from the WebSphere Administrative Console.

5. Enter the following URLs to run the snoop servlet:

```
http://<web_server_hostname.com>/servlet/snoop
https://<web_server_hostname.com>/servlet/snoop
```

3.8.7 Modify was.deployed.EJB.xml (workaround)

This procedures is only required if you plan on later configuring horizontal cloning.

Before creating an instance, back up and edit the /opt/WebSphere/CommerceServer/xml/config/was.deployed.EJB.xml file.

From:

```
<enterprise-application action="create" name="WebSphere Commerce Enterprise
Application - $DRIVER_INSTANCE_NAME$">
```

To:

```
<enterprise-application action="create" name="WC Enterprise App
$DRIVER_INSTANCE_NAME$">
```

Note: Considerations for creating an instance on a shared file system from the start:

- Specify a central directory for the instance's root path. This way the configuration files can be easily shared by application server clones on multiple server machines. If specifying a directory other than the default, then this directory must already exist, and the wasuser must have write access there.
- Change the directory for the profile path (in the Payment Manager panel).
- Logs (tracefile location and message file location) and cache directories should be local to each server.

3.8.8 Update setenv.sh (workaround)

In order to work around a MassLoader property `file not found` error when WebSphere Application Server V4 FixPak 3 is installed, update the setenv.sh file.

If you do not update the setenv.sh, you will see the following error message:

```
002.07.29 09:54:17.397 java.lang.Class event   MassLoader
LOG0040E The message file
com.ibm.wca.MassLoader.Validator.ValidatorProperty was not found.  The
message key is ValidatorReceivedEvent ,
p1=com.ibm.wca.MassLoader.Events.RecordEvent
```

To work around this problem, back up and update the setenv.sh file found in the /opt/WebSphere/CommerceServer/bin directory. Search for export JAVA_EXE and append the Xbootclasspath as follows:

Syntax:

```
export JAVA_EXE="$JAVA_HOME/bin/java -Xbootclasspath/p:<xxx>”
```

Where <xxx> is the full path of the `jlog.jar` file.

Example:

```
export JAVA_EXE="$JAVA_HOME/bin/java
-Xbootclasspath/p:/opt/WebSphere/CommerceServer/lib/jlog.jar"
```

3.8.9 Change file ownership and permissions (workaround)

After installing the WebSphere Commerce FixPak V5.4.0.3 and before instance creation, you will need to manully update the ownership of some files in the <wc_home>/bin and <wc_home>/schema directories:

1. Start an Linux console window and log on as root.

2. Change ownership and file permissions.

   This can be accomplished by running wcb.sh or by using the **chown** and **chmod** commands. The files that need to be updated to have the non-root user (wasuser) the owner are in the following directories:

   – /opt/WebSphere/CommerceServer/bin
   – /opt/WebSphere/CommerceServer/schema/db2

   We ran the following command to address this issue (refer to 3.8.2, “Post wcb.sh configuration” on page 92 for details):

   ```
   # /opt/WebSphere/CommerceServer/bin/wcb.sh
   ```

> **Note:** If you do not update the ownership and permissions on these files to match the other files, you will see the following error in the WebSphere Administrative Console window:
>
> ```
> “Error : DB2 : serviceDB : ioe :
> /opt/WebSphere/CommerceServer/bin/servicedb.db2.sh (The file permissions
> do not allow the specified action).
> ```

3.9 WebSphere Commerce instance creation

This section describes the high-level steps necessary to create a WebSphere Commerce instance using the Configuration Manager for the Linux.

Full instructions are provided for creating or modifying a WebSphere Commerce instance in the *Installation Guide, IBM WebSphere Commerce V5.4 Professional and Business Edition for Linux* product guide.

This section includes the following topics:

- Prerequisites for WebSphere Commerce instance creation
- Start the Configuration Manager Server
- Start the Configuration Manager (client)
- Create an instance with the Instance Creation Wizard
- Verify the WebSphere Commerce instance creation

3.9.1 Prerequisites for WebSphere Commerce instance creation

This section describes the prerequisites needed before creating an instance.

- Ensure the steps listed in 3.8, “WebSphere Commerce post-install configuration” on page 89 have been completed prior to instance creation.
- Ensure the following application servers are running:
  - DB2 Server

    ```
    # su - db2inst2
    # db2start
    ```
  - WebSphere Application Server Administrative Server

    ```
    # su - wasuser
    $ /opt/WebSphere/AppServer/bin/startupServer.sh &
    $ tail -f /opt/WebSphere/AppServer/logs/tracefile
    ```
- Ensure that you have X-Windows access and that the TERM and DISPLAY environment variables have been set appropriately.
- To create an instance, you will require over 120 MB of free disk space in the /opt/WebSphere/AppServer/installedApps directory, and at least an equal amount in /tmp.

3.9.2 Start the Configuration Manager Server

To start the Configuration Manager Server, do the following:

1. Log on as the non-root user, wasuser:

   ```
   # su - wasuser
   ```
2. Start the Configuration Manager Server:

   ```
   $ cd /opt/WebSphere/CommerceServer/bin
   $ ./config_server.sh
   ```

   You should see the following message if started successfully:

   ```
   Registry created.
   CMServer bound in registry.
   ```

   Note: The Configuration Manager Server should be stopped after you exit Configuration Manager (client). Make sure that the server process is not left running in the background.

3.9.3 Start the Configuration Manager (client)

After Configuration Manager Server is started, start the Configuration Manager (client) as follows:

1. Log on as the non-root user, wasuser:

   ```
   # su - wasuser
   ```

2. Start the Configuration Manager (client) as follows:

   ```
   $ cd /opt/WebSphere/CommerceServer/bin
   $ ./config_client.sh &
   ```

3. Log on to the Configuration Manager.

   – User ID: `webadmin`
   – Password: `webibm (default)`

   You will be prompted to reset the default password.

   > **Note:** The new password must be a minimum of eight characters long, and contain at least one numeric digit and one alphabetical character. It also cannot be the same as the user name, or contain more than four consecutive occurrences of the same character.

3.9.4 Create an instance with the Instance Creation Wizard

To create an instance using the Configuration Manager Instance Creation Wizard, do the following:

1. From the Configuration Manager, select and expand **<your_hostname> -> Instance List**, and then right-click **Create instance**.

2. We accepted the default values unless otherwise noted in the following list of settings we used to create a WebSphere Commerce instance for our scenario:

   Instance tab:

   – Instance name: `wc1`
   – Instance root path: `/opt/WebSphere/CommerceServer/instances/wc1`
   – Merchant key: `<16_character_hex_value>`

   Database tab*:*

   – Database administrator name: `db2inst1`
   – Database administrator password: `<password>`
   – Database administrator home directory: `/home/db2inst1`
   – Database name: `wc1db`
   – Database type: `DB2`
   – Database user name: `db2inst1`

- Database user password: `<password>`
- Database user home directory: `/home/db2inst1`
- Select **Run Database Performance Wizard**
- Select **Set as active Database**

Languages tab: (Default English)

Web Server tab:

- Hostname: `comlx1.itso.ral.ibm.com`
- Web Server Type: `IBM HTTP Server`
- Primary Document Root: `/opt/HTTPServer/htdocs/en_US`
- Server Port: `80`
- Authentication Mode: `Basic`

WebSphere tab:

- DataSource Name: `WebSphere Commerce DB2 DataSource wc1`
- Port Number: `2222`

  The port number should match what was configured during the post install script (wcb.sh) to run the WebSphere Administrative Server as a non-root user.
- We accepted the defaults for the remaining options.

Payment Manager tab:

- Hostname: `comlx1.itso.ral.ibm.com`
- We accepted the defaults for the remaining options

Log system tab: (we accepted the default settings)

Messaging tab: (we accepted the default settings)

Auction tab: (we accepted the default settings, not enabled)

3. When all the information has been entered, click **Finish** to create the WebSphere Commerce instance.

> **Note:** The WebSphere Commerce instance creation takes approximately 40 minutes with the options selected and hardware used.
>
> Notice the progress indicator at the bottom of the Configuration Manager window.

4. When the instance creation is complete, you should see a message: `Instance was successfully created`. Click **OK**.
5. Click **License User Management** in the left panel.
6. Enter the appropriate store license number and then click **Apply**.

7. Close the Configuration Manager.

> **Security note:** After closing the Configuration Manager, stop the Configuration Manager Server (config_server.sh) for security purposes.

3.9.5 Verify the WebSphere Commerce instance creation

To verify that the WebSphere Commerce instance has been created properly, complete the following steps:

1. Check the following configuration and log files:

   – <instance>.xml: This file contains all of the configuration information about your WebSphere Commerce instance.

     ```
     /opt/WebSphere/CommerceServer/instances/<instance>/xml/<instance>.xml
     ```

   – createdb.log: This file contains information about the WebSphere Commerce database creation.

     ```
     /opt/WebSphere/CommerceServer/instances/<instance>/logs/createdb.log
     ```

   – populatedb.log: This file contains information about the WebSphere Commerce database population process.

     ```
     /opt/WebSphere/CommerceServer/instances/<instance>/logs/populatedb.log
     ```

   – WASConfig.log: This file contains information about the enterprise application deployment and configuration of your WebSphere Commerce instance (application server), within WebSphere Application Server.

     ```
     /opt/WebSphere/CommerceServer/instances/<instance>/logs/WASConfig.log
     ```

   – <instance>.log: This file describes the operation of the WebSphere Commerce application server. Use this log to ensure that the server has started correctly.

     ```
     /opt/WebSphere/CommerceServer/instances/<instance>/logs/<instance>.log
     ```

2. Check that the database tables have been created correctly.

> **Note:** Some connection errors for the cache daemon may be logged in the wcs.log after the instance automatically starts up. This is because the remote Web server has not yet been configured to receive servlet requests for WebSphere Commerce (plugin configuration needs to be refreshed).

3.10 WebSphere Commerce post-instance config

This section describes the steps that need to be completed for the WebSphere Commerce configuration after instance creation, including the following tasks:

- Regenerate the Web server plugin
- Stop/start the WebSphere Commerce instance
- Compile the WebSphere Commerce tools JSPs (WCS Tools)
- Rebind data source (workaround)

3.10.1 Regenerate the Web server plugin

There are two methods of regenerating the plugin, via the command line or using the WebSphere Administrative Console. Whenever new applications or virtual host definitions are added or changed for the WebSphere Application Server, it is necessary to refresh the plugin configuration file (plugin-cfg.xml). Now that the WebSphere Commerce application server has been deployed and new host aliases added, we must regenerate the Web server plugin.

To regenerate the plugin from the WebSphere Application Server Administrative Console, do the following:

1. After the WebSphere Commerce - <instance> application server has been deployed (via Configuration Manager Instance Creation Wizard), you will need to refresh the WebSphere Administrative Console if running to see the new application server. The WebSphere Administrative Console can be refreshed by selecting the node and then clicking the refresh icon from the toolbar.
2. Select and expand **WebSphere Administrative Domain -> Nodes.**
3. Right-click the **<your_node>** and select **Regen Webserver Plugin**. The updated plugin-cfg.xml is written to the /opt/WebSphere/AppServer/config directory.
4. Exit the WebSphere Application Server Administration Console.

Tip: To regenerate the Web server plugin (plugin-cfg.xml) from the command line as an alternative to using the WebSphere Application Server Administrative Console, do the following:

```
# cd /opt/WebSphere/AppServer/bin
# ./GenPluginCfg.sh -adminNodeName <node_name> -nameServicePort
<port_#>
```

For example:

```
# ./GenPluginCfg.sh -adminNodeName comlx1 -nameServicePort 2222
```

This utility is only present if the Server component was selected during the installation of WebSphere Application Server.

It does not seem to matter whether GenPluginCfg.sh is run as root or wasuser-- the updated plugin-cfg.xml is still owned by wasuser.

5. Restart the IBM HTTP Server.

3.10.2 Stop/start the WebSphere Commerce instance

Once the WebSphere Commerce instance has been created, it will be started automatically.

Note: Within our test environment, after the WebSphere Commerce instance application server was deployed, it was not visible from the WebSphere Application Server Administrative Console. We closed the Administrative Console and restarted it and it then appeared. Alternatively, you can aslo refresh the WebSphere Administrative Console by selecting the node and then clicking the refresh icon from the toolbar.

If you modify your instance at a later time, you will need to restart it. You can stop and start your instance with the WebSphere Application Server Administrative Console by completing the following steps:

1. Start the WebSphere Application Server Administration Console.

   Ensure that you have started the Admin Server and Administrative Console as a non-root user if you selected this option during wcb.sh configuration.

   Refer to 3.8.4, “Start WebSphere Administrative Console (non-root wasuser)” on page 95 for details.

2. Select and expand **WebSphere Administrative Domain -> Nodes -> <your_hostname> -> Application Servers.**

3. Select **WebSphere Commerce Server - <instance>**, and right-click **Start** or **Stop** as appropriate.

3.10.3 Compile the WebSphere Commerce tools JSPs (WCS Tools)

We recommend that you pre-compile the tools JSPs for the WebSphere Commerce. Pre-compiling the JSPs will significantly reduce the amount of time needed to load the WebSphere Commerce tools. By default, the WebSphere Application Server will compile the JSPs the first time the JSP is requested.

To batch compile JSPs for the WebSphere Commerce tools, do the following:

1. Ensure that the WebSphere Application Server is started. If it has not been started already, then open a Solaris console and enter the following commands:

   ```
   # su - wasuser
   $ /opt/WebSphere/AppServer/bin/startupServer.sh &
   ```

   To monitor when the WebSphere Application Server is finally started, type:

   ```
   % tail -f /opt/WebSphere/AppServer/tracefile
   ```

   Tail the tracefile until the `adminServer open for e-business` message appears.

2. From a command prompt, go to CommerceServer directory and run the script:

   ```
   $ cd /opt/WebSphere/CommerceServer/bin
   ```

   Syntax:

   ```
   $ ./WCSJspBatchCompiler.sh -enterpriseApp "WC Enterprise App <instance>"
   -webModule "WCS Tools" -nameServerHost "<hostname>" -nameServerPort
   <port_number>
   ```

   – `<instance>` is the name of your WebSphere Commerce instance (for example, wc1.

   – `<hostname>` is the name of your WebSphere Commerce server machine (for example, comlx1)

   – `<port_number>` is the port specified in running the wcpostinstall.sh script (for example, 2222).

Note: Take note of the -enterpriseApp name "WC Enterprise App <instance>" name for our example. Prior to creating our instance, we modified the /opt/WebSphere/CommerceServer/xml/config/was.deployed.EJB.xml file (see 3.8.7, "Modify was.deployed.EJB.xml (workaround)" on page 98):

From: `WebSphere Commerce Enterprise Application - <instance>`

To: `WC Enterprise App <instance>`

This was done to address a horizontal cloning problem we discovered and created a workaround for. The -enterpriseApp name needs to be reflected in the command line arguments provided to WCSJspBatchCompiler.sh.

Example:

```
$ ./WCSJspBatchCompiler.sh -enterpriseApp "WC Enterprise App wc1"
-webModule "WCS Tools" -nameServerHost "comlx1" -nameServerPort 2222
```

The compilation of the JSPs takes several minutes to complete for the WebSphere Commerce tools.

3. If successful, you should see messages scrolling by, notifying you of the compile and the name of the JSPs.

Note: We recommend that you create a script to perform the pre-compile. The script will need to be run anytime the cache is flushed for the application server.

For example:

- Create a script wctoolsjsp.sh in the /opt/WebSphere/CommerceServer/bin directory like the following:

```
#!/bin/sh
./WCSJspBatchCompiler.sh -enterpriseApp "WC Enterprise App wc1"
-webModule "WCS Tools" -nameServerHost "comlx1" -nameServerPort 2222
```

- Execute the wctoolsjsp.sh script while logged in as wasuser:

```
# su - wasuser
$ cd /opt/WebSphere/CommerceServer/bin
$ chmod 755 wctoolsjsp.sh
$ ./wctoolsjsp.sh
```

3.10.4 Rebind data source (workaround)

When accessing a WebSphere Commerce tools pages (for example, Commerce Accelerator, Store Services, and Admin Console), with WebSphere Application Server V4 FixPak 4 (or 3) installed, a blank page will be generated and the log file will state the following:

```
Error detected when initializing "EJB property”.
```

To work around this problem, do the following:

1. Log on on as the non-root user (wasuser):

   ```
   # su - wasuser
   ```

2. We will need to run the WebSphere Control Program (wscp) as a non-root user. In order to do so, we will need to create a .wscprc properties file with the following entries so that we can connect to the WebSphere Administrative Server on port 2222 in a similar fashion to what is done for the adminclient.sh.

   Create the .wscprc file in the root of the wasuser home directory (for example, /home/wasuser) with the entries seen in Example 3-4:

Example 3-4 wscp properties file (.wscprc) to connect as non-root user

```
wscp.hostName=comlx1.itso.ral.ibm.com
wscp.hostPort=2222
```

   where <comlx1.itso.ral.ibm.com> is the fully qualified hostname of the WebSphere Administrative Server, and

   where 2222 is the port number for the WebSphere Administrative Server running as non-root.

3. Change to the <WAS_HOME>/AppServer/bin directory:

   ```
   $ cd /opt/WebSphere/AppServer/bin
   ```

4. Enter the following command to rebind the datasource

   ```
   $ ./wscp.sh -f rebindDataSources.tcl
   ```

   You should see messages like the following:

   ```
   Rebinding all DataSources ...
   Finished rebinding DataSources
   ```

5. Restart the WebSphere Commerce - <instance> application server.

> **Note:** For more information on the WebSphere Control Program (wscp), refer to the *IBM WebSphere V4.0 Advanced Edition Handbook*, SG24-6176.

3.11 WebSphere Commerce Payments configuration

This section describes the high-level steps required to configure WebSphere Commerce Payments V3.1.3 for use with WebSphere Commerce V5.4.

3.11.1 Start the WebSphere Commerce Payments Server

To start the WebSphere Commerce Payments Server, do the following:

1. Ensure the following servers are started:
   - IBM DB2
   - IBM HTTP Server
   - WebSphere Administrative Server
2. From the WebSphere Administrative Console, select **WebSphere Administrative Domain -> Nodes -> <your_node> -> Application Servers -> WebSphere Commerce Payments**. Right-click **Start**.
3. Start a Linux Console window and change to the WebSphere Commerce Payments home directory.

   ```
   # cd /opt/PaymentManager
   ```

4. Start the WebSphere Commerce Payments Server as follows:

   ```
   # ./IBMPayServer
   Password: <your_db2_password>
   ```

   You should see a message like the following if started successfully:

   ```
   IBM WebSphere Commerce Payments has started successfully.
   ```

3.11.2 Add payment information for the store

In order for the store to function correctly once it is published, and for the shoppers to be able to place orders with the store, you need to add payment information to the store.

> **Note:** This is not necessary if you plan to publish one of the sample stores when WebSphere Commerce Payments is running. The sample stores have included WebSphere Commerce Payments information.

3.12 Deploy a store

In order to verify the runtime environment for WebSphere Commerce, we need to publish a store. This can be a store you have created or a sample store. In this example, we publish the ToolTech sample store.

3.12.1 Download updated model stores from the Web

The WebSphere Commerce store models shipped with the product have been updated and made available for download.

1. To download the updated IBM WebSphere Commerce V5.4, Business Edition sample stores zip and readme, enter the following URL:

   `http://www-1.ibm.com/support/manager.wss?rs=494&rt=0&org=SW&doc=4001259`

   In our example, we downloaded the Linux zip and readme.

2. Refer to the ModelsUpdateReadme.pdf for instructions on how to install the updated model stores.

3.12.2 Publish the store from Store Services

To create and publish a store based on the ToolTech sample store from Store Services, do the following:

1. Make sure the following are started:
   - IBM DB2
   - IBM HTTP Server
   - IBM WebSphere Administrative Server (running as non-root)
   - WebSphere Commerce Server - <instance_name> (WebSphere Application Server)
   - WebSphere Commerce Payments (WebSphere Application Server)
   - IBMPayServer (command executed and running)

2. Start Store Services by entering the following URL in a Web browser:

   ```
   https://<fully_qualified_hostname>:8000/storeservices
   ```

   > **Note:** The WebSphere Commerce Web browser-based tools requires Microsoft Internet Explorer 5.5 or higher, which is only available on Windows.

3. Log on using the “wcsadmin” user and password (the default password is wcsadmin).

Tip: In the event that you incorrectly enter your password and the account becomes disabled, do the following:

- Open a DB2 command line processer window.
- Connect to the WebSphere Commerce instance database:

  ```
  db2 => connect to wc1db
  ```

- Enter the following DB2 command to reset the WebSphere Commerce wcsadmin logon ID:

  ```
  db2 => update USERREG SET STATUS=1, PASSWORDRETRIES=0 WHERE
  LOGONID='wcsadmin'
  ```

4. Select the sample store that you want your new store to be based on. For example, we selected the following for ToolTech and then clicked **OK**:
   - Select **ToolTech_en_US_es_ES.sar**
   - Store archive: `tooltech`
   - Store directory: `tooltech`
   - Store owner: select **Seller Organization**
5. Your should see the message `tooltech.sar created successfully`. Click **OK**.
6. To publish the store, select the store you created (for example, we selected tooltech.sar) and then click **Publish**.
7. We accepted the defaults for the remaining options.

   The publishing of a store can take approximately one-half to one hour, depending on your system specifications. This can take longer if you have created your own store and included many products.

Security note: When you want to close Store Services, you should use the Logout link in the top right-hand corner of the Store Services window before actually closing the browser, so as to avoid a security risk.

8. The publishing state is reflected in the Publish status column. Click **Refresh** to update the status.
9. Select the store archive from the list and click **Publish Summary** to see the results of the publish.

10. To review the publishing logs in the event of a failure, check the following:

```
<wc_home>/instances/<instance>/logs/messages.txt
<wc_home>/instances/<instance>/logs/trace
<wc_home>/instances/<instance>/logs/ecmsg_hostname_timestamp.log
<wc_home>/instances/<instance>/logs/wcs.log
```

11. To perform a basic verification test of the store after publishing is finished, click **Publish Summary -> Launch Store.** Add the store to your favorites or bookmarks list.

    You will notice that the store JSPs are slow to load the first time they are accessed. Once the WebSphere Application Server has compiled them, subsequent accesses will be much faster. To address this issue, we will precompile all of the store JSPs in a later step.

3.12.3 Compile the WebSphere Commerce tools JSPs (WCS Tools)

We recommend that you pre-compile the store JSPs for the WebSphere Commerce. Refer to 3.10.3, “Compile the WebSphere Commerce tools JSPs (WCS Tools)” on page 107 for more detailed information.

Example store JSPs:

```
> WCSJSPBatchCompiler -enterpriseApp “WC Enterprise App wc1” -webModule
“WCS Stores” -nameServerHost “comlx1” -nameServerPort 2222
```

> **Note:** If there are errors in the command prompt window during the compilation of the JSPs, you can safely ignore them most of the time.

3.13 WebSphere Commerce runtime verification

Now that the WebSphere Commerce runtime environment is configured and a store is published, you need to verify that all the components are functioning correctly.

3.13.1 Verify the WebSphere Commerce administration tools

To verify the WebSphere Commerce administration tools are working properly, do the following:

1. Ensure the following Windows services are started:

   – IBM DB2
   – IBM HTTP Server
   – WebSphere Administrative Server (run as non-root)

2. Ensure that the following application servers are started from the WebSphere Application Server Administrative Console:

   – WebSphere Commerce Server - <instance name>
   – WebSphere Commerce Payments

3. Ensure the IBMPayServer is started.

4. Start the Store Services.

   ```
   https://<hostname>:8000/storeservices
   ```

5. Start the WebSphere Commerce Administration Console:

   ```
   https://<hostname>:8000/adminconsole
   ```

6. Start the WebSphere Commerce Accelerator:

   ```
   https://<hostname>:8000/accelerator
   ```

3.13.2 Verify the WebSphere Commerce store

To verify the WebSphere Commerce store is working properly, do the following:

1. Precompile the store JSPs.

   In order to view the store pages correctly without waiting for the WebSphere Application Server to compile each page after it is requested, we first need to compile all the sample store JSPs. In our example, we have already precompiled the JSPs.

2. Enter the store home page URL. If you do not recall the URL, it can be accessed from the Publish Summary of Store Services. The store home page URL will look something like the following:

   ```
   http://<hostname>/webapp/wcs/stores/servlet/<store>/index.jsp
   ```

   For example - tooltech:

   ```
   http://comlx1.itso.ral.ibm.com/webapp/wcs/stores/servlet/tooltech/index.jsp
   ```

3. Register a user.

4. Browse the store catalog.

5. Select an item to purchase.

6. Complete an order by selecting a valid payment type.

7. Enter 16 zeros (0s) as your credit card number so that you are able to test it.

3.13.3 Verify WebSphere Commerce Payments functionality

Once you get the order confirmation within the store, do the following to verify the integration of WebSphere Commerce and WebSphere Commerce Payments:

1. Log on to the WebSphere Commerce Payments Logon window using the wcsadmin account.

   ```
   http://<hostname>/webapp/PaymentManager
   ```

2. Click **Approve**. You should see the order that was just placed. Select the order and click **Approve**.
3. You should see a message indicating that the order was approved and that there are no more pending orders to approve.

3.14 Auto start servers

After the installation, you may optionally configure the server components to start automatically upon the Linux system reboot. To automate the startup of servers such as DB2, WebSphere Application Server, IBM HTTP Server, WebSphere Commerce, and WebSphere Commerce Payments, refer to Appendix B, “Automating server startup” on page 161.

4

Implementing a 2-tier remote DB2 server runtime

The scenario highlighted in this chapter describes how to implement a 2-tier remote DB2 server runtime environment for IBM WebSphere Commerce V5.4, Business Edition for Linux.

In this scenario, the IBM HTTP Server, WebSphere Application Server, WebSphere Commerce, WebSphere Commerce Payments and DB2 Client are installed on node B. The DB2 Server is installed on node C.

4.1 Planning for the remote DB2 server runtime

In this scenario, we guide the user through the process of installing, configuring, and verifying each component of the desired two-tier remote DB2 runtime environment. This includes modifications made for security, systems management, and scalability. Figure 4-1 depicts the components of the runtime.

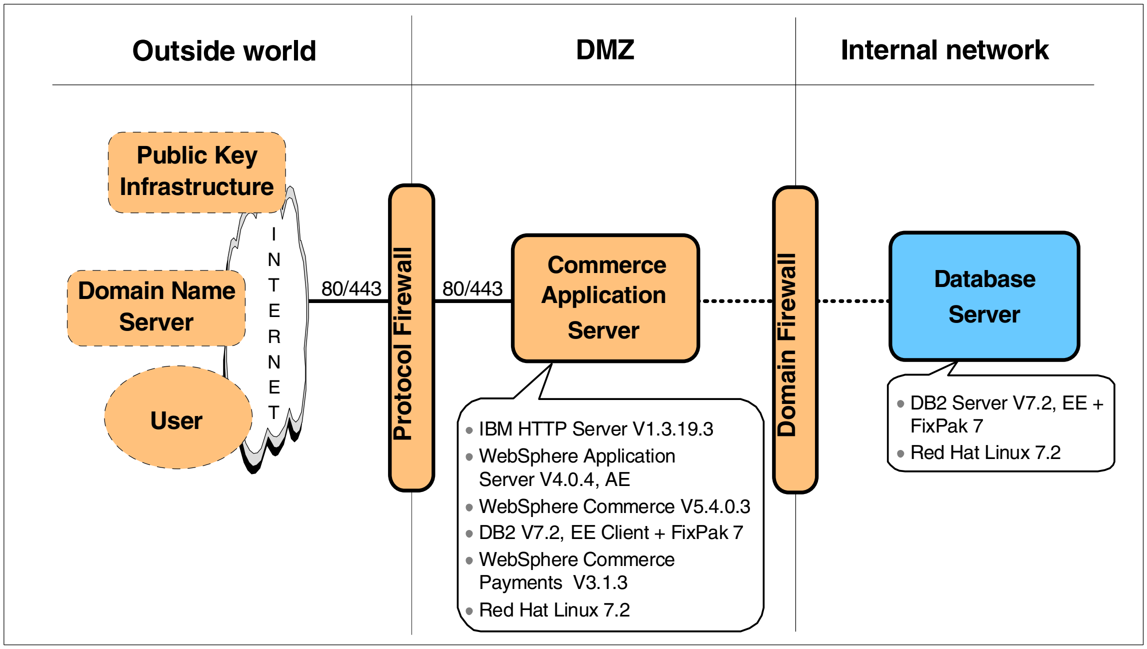

Outside world
DMZ
Internal network
Public Key Infrastructure
Domain Name Server
User
INTERNET
80/443
Protocol Firewall
80/443
Commerce Application Server
Domain Firewall
Database Server
IBM HTTP Server V1.3.19.3
WebSphere Application Server V4.0.4, AE
WebSphere Commerce V5.4.0.3
DB2 V7.2, EE Client + FixPak 7
WebSphere Commerce Payments V3.1.3
Red Hat Linux 7.2
DB2 Server V7.2, EE + FixPak 7
Red Hat Linux 7.2

Figure 4-1 Product mapping for the two-tier remote DB2 server runtime for Linux

The procedure to install a 2-tier remote DB2 server runtime environment is similar to that for the single-tier installation. The primary difference is the use of a DB2 client on the commerce application server node and the DB2 client being configured to communicate with a remote DB2 server node.

This section is organized as follows:

- Red Hat Linux installation
- DB2 Server installation
- DB2 Client installation
- IBM HTTP Server and WebSphere plugin installation
- WebSphere Application Server installation
- WebSphere Commerce installation
- WebSphere Commerce Payments installation
- WebSphere Commerce post-install configuration

- WebSphere Commerce instance creation
- WebSphere Commerce post-instance config
- WebSphere Commerce Payments configuration
- WebSphere Commerce runtime verification

In our test environment, the remote DB2 server hostname is `db2lx1`, and the commerce application server hostname is `comlx1`.

Refer to 3.1, “Planning” on page 33 for more detailed information on the sofware and hardware requirements as well as the software and hardware planning information used within the ITSO test environment.

4.2 Red Hat Linux installation

Refer to 3.2, “Red Hat Linux installation” on page 36 for details on the installation and configuration of Linux in preparation for WebSphere Commerce and supporting software components.

4.3 DB2 Server installation

In this scenario, the DB2 Server was installed on a separate node (hostname db2lx1). Refer to 3.3, “DB2 Server installation” on page 39 for details on the installation and configuration of the DB2 Server.

Note: In this example, the DB2 server node name is db2lx1. When refering to the detailed DB2 installation instructions in 3.3, “DB2 Server installation” on page 39, make sure you use the db2lx1 node name (not comlx1). In this example, the following procedure is performed on the remote DB2 server.

1. Catalog the TCP/IP node.

   In order to access the administration database via TCP/IP, the DB2 node must first be cataloged.

   ```
   $ db2 catalog tcpip node db2lx1 remote db2lx1 server db2cdb2inst1
   ```

2. Verify the attach to the TCP/IP node:

   ```
   $ db2 attach to db2lx1 user db2inst1 using db2inst1
   ```

3. Catalog the database.

   The administration database must now be catalogued as part of this TCP/IP node:

   ```
   $ db2 catalog db was4 as was40 at node db2lx1
   ```

4.4 DB2 Client installation

This section describes how to install and configure the DB2 Client on the commerce application server node. The DB2 Client will be configured to communicate to the DB2 Server where the WebSphere repository database, WebSphere Commerce instance/store database, and WebSphere Commerce Payments database will reside.

The steps for installing the DB2 Client software are very similar to those of the DB2 Server installation, but they are outlined here for clarity.

The section is organized into the following tasks:

- DB2 pre-installation tasks
- Install the DB2 Client
- Verify the DB2 Client installation
- Install DB2 UDB V7 FixPak 7
- Configure and verify the DB2 Client
- Configure and verify the DB2 client and server connectivity

4.4.1 DB2 pre-installation tasks

Prior to installing the DB2 Client the following check needs to be completed:

- Verify that there are no existing active services that use the same DB2 TCP/IP ports on the server:
  - 523 (DB2 Server)
  - 50000 (DB2 instance connection port)
  - 50001 (DB2 instance interrupt port)
  - 50002 (DB2 Control Server)

  We suggest using the following command for this task:

  ```
  netstat -an | grep LISTEN
  ```

4.4.2 Install the DB2 Client

In order to install IBM DB2 Universal Database V7.2, Enterprise Edition for Linux, perform the following steps:

1. Log in as root.
2. Start a terminal session.
3. Mount the DB2 V7.2 CD-ROM:

   ```
   # mount -r /dev/cdrom /mnt/cdrom
   # cd /mnt/cdrom
   ```

> **Note:** If you are running X Windows with either the KDE or Gnome window managers, the CD-ROM may automatically be mounted for you. To verify this, simply perform a `mount | grep /mnt/cdrom` and if you get any output then the CD-ROM has already been mounted.

4. Start the DB2 installer program:

```
# ./db2setup
```

5. In the Install DB2 V7.2 window, select only the following options:
   - Select **DB2 Administration Client**
   - Select **DB2 Application Development Client**

     This option is required to load stored procedures needed for WebSphere Commerce.

> **Navigation tips:**
>
> ► Press **Tab** to move between available options and fields.
>
> ► Press **Ctrl+L** to refresh the window at any point.
>
> ► Select and press **Enter** to select an option.

6. Select the DB2 Product Library's **Customize** option and press **Enter**.
7. In the DB2 Product Library window, highlight the appropriate option for your locale under the DB2 Product Library (HTML) section, then highlight **OK** and press **Enter**.
8. Select **OK** and press **Enter**.
9. In the Create DB2 Services window, select the **Create a DB2 Instance** option, highlight **OK** and press **Enter**.
10. When the DB2 instance authentication window appears, enter the following:
    - User Name: `<db2_instance_owner>`
    - User ID: `<use default UID>`
    - Group Name: `db2iadm1`
    - Group ID: `<use default GID>`
    - Home Directory: `/home/<db2_instance_owner>`
    - Password: `<user_password>`
    - Verify Password: `<user_password>`

Note: The DB2 installer uses the above information to automatically perform the following operations:

- Create a group db2iadm1
- Create a user <db2_instance_owner> with primary group db2iadm1
- Sets the <db2_instance_owner> password to the <user_password> value. The password used must meet DB2 requirements: eight characters or fewer and not containing the characters “<“ or “>”.

It also changes the ownership (owner:group) of the /home/<db2_instance_owner> directory to be <db2_instance_owner>:db2iadm1

11. Select **OK** and press **Enter**.
12. Again, highlight **OK** and press **Enter**.
13. The Summary Report window is displayed, listing the product components to be installed. Select **Continue** and press **Enter**.
14. A warning window appears, indicating this is your last chance to stop. Select **OK** and press **Enter**.

    The db2setup program installs the selected components. Depending on the speed of your processor, this can take up to 15 minutes.
15. You may be prompted to register the product. Complete the registration then go back to the install window.
16. When the install completes, a notice window informs you of whether the installation was successful. Select **OK** and press **Enter**.
17. Scan the Status Report to ensure that all components were installed successfully. Select **OK** and press **Enter**.
18. In the DB2 Installer window, highlight **Close** and press **Enter**.
19. A window appears asking `Do you want to exit the DB2 Installer?` Select **OK** and press **Enter**.

    The DB2 installation is now complete.
20. Unmount the CD-ROM:

```
# cd /
# umount /mnt/cdrom
```

4.4.3 Verify the DB2 Client installation

To verify the DB2 Server installation, complete the following tasks:

- Check home directory permissions.
- Check DB2 instance owner profile.
- Check DB2 instance symbolic links.
- Check DB2 release level.

Check home directory permissions

Check that the home directory ownership has been correctly set up by the db2setup program.

Table 4-1 DB2 home directory required permissions

Home directory path	Owner	Group	Permissions[a]
/home/<db2_instance_owner>	<db2_instance_owner>	db2iadm1	drwxr-xr-x

[a] The permissions need to be such that the Owner can read, write, and execute files and directories within the path. Group members' and other users' access rights are up to each company's security policies and business needs.

If a DB2 related home directory has not been correctly configured, perform the following steps:

1. Log in as root and start a terminal session.
2. Issue the following command, substituting values from Table 4-1:

```
# chown -fR <owner>:<group> <home_directory_path>
```

Check DB2 instance owner profile

The DB2 Server installation should set up the .bashrc environment file of the <db2_instance_owner> so that the DB2 environment is set up when the user logs in.

The following content should have been added to the file:

```
if [ -f ~/sqllib/db2profile ] ; then
. ~/sqllib/db2profile
fi
```

If not present, manually edit the file to add the above content.

Check DB2 instance symbolic links

The DB2 Server installation automatically creates a DB2 instance <db2_instance_owner> under the /home/<db2_instance_owner> directory. As part of the instance creation, db2setup should create symbolic links in the /home/<db2_instance_owner>/sqllib directory to files under /usr/IBMdb2/V7.1.

Perform the following steps to check whether the symbolic links have been created:

1. Log in as root, and start a terminal session.
2. Change the directory to /home/<db2_instance_owner>/sqllib.
3. Check whether a number of symbolic links exist pointing to files under /usr/IBMdb2/V7.1.
4. If not, issue the following commands:

```
# cd /usr/IBMdb2/V7.1/cfg
# ./db2ln
```

Check DB2 release level

Check that DB2 has the correct internal release level to meet WebSphere Application Server requirements:

1. Change to user <db2_instance_owner>:

```
# su - <db2_instance_owner>
```

2. Enter the following command:

```
# db2level
```

   An internal release level of 7.1.0.60 (or whatever level you have installed) should be indicated.

4.4.4 Install DB2 UDB V7 FixPak 7

For detailed information, refer to 3.3.4, “Install DB2 UDB V7 FixPak 7” on page 49.

4.4.5 Configure and verify the DB2 Client

After the DB2 Client installation, a number of configuration tasks must be performed.

The following tasks need to be completed to configure the DB2 client:

- Update root administrative groups
- Update JDBC level to JDBC2
- Verify DB2 environment
- Update root environment file

Update root administrative groups

> **Note:** If only DB2 Client is installed on the system, this step is not needed.

The DB2 Server installation should add the db2asgrp administrative group to the root user.

Perform the following steps to check whether the root account's administrative groups have been amended:

1. Log in as root and start a terminal session.
2. Issue the following command for groups:

   ```
   # groups
   ```

3. If db2asgrp is not listed as one of the groups assigned to root, you can use the Red Hat GUI tools to reconfigure the root user or you can enter the following:

   ```
   # vigr
   ```

   This will bring you into a vi editing session with a locked version of the /etc/group file. Find the line in the file that starts with `db2asgrp:` and add the user, root, to the end of it (separated by a comma from other user names). The line should look similar to this after editing:

   ```
   db2asgrp:db2as,root
   ```

   Once you have added that, press **ESC+:wq** to save and quit vi. The system will then prompt you to edit the shadow group file. Choose **Yes** and do the same thing to that file, saving and quitting at the end of your editing.

Update JDBC level to JDBC2

IBM WebSphere Application Server 4.0 requires the use of JDBC2.0, whereas the default installation of IBM DB2 V7.2 uses JDBC1.2. To update the DB2 JDBC level, complete the following steps:

1. Change to user <db2_instance_owner>:

   ```
   # su - <db2_instance_owner>
   ```

2. Add the following content to the end of the <db2_instance_owner> .bashrc environment file:

   ```
   if [ -f ~/sqllib/java12/usejdbc2 ] ; then
   . ~/sqllib/java12/usejdbc2
   fi
   ```

Verify DB2 environment

After the above configuration steps, we need to check that the environment being set up by the db2profile and usejdbc2 scripts is correct:

1. Change to user <db2_instance_owner>:

   ```
   # su - <db2_instance_owner>
   ```

2. Issue the following command:

   ```
   $ set | grep [Dd][Bb]2
   ```

3. Check that the environment variables in this output match the values in Table 4-2.

Table 4-2 DB2 Server required environment variables

Environment variable	Required value
DB2COMM	tcpip
DB2DIR	/usr/IBMdb2/V7.1
DB2INSTANCE	<db2_instance_owner>
INSTHOME	/home/<db2_instance_owner>
LD_LIBRARY_PATH	:/home/<db2_instance_owner>/sqllib/lib
LIBPATH	..........:/home/<db2_instance_owner>/sqllib/java12:/home/<db2_instance_owner>/sqllib/lib
CLASSPATH	/home/<db2_instance_owner>/sqllib/function:/home/<db2_instance_owner>/sqllib/java12/db2java.zip:/home/<db2_instance_owner>/sqllib/java/runtime.zip
PATH	..........:/home/<db2_instance_owner>/sqllib/java12:/home/<db2_instance_owner>/sqllib/bin:/home/<db2_instance_owner>/sqllib/adm:/home/<db2_instance_owner>/sqllib/misc

Update root environment file

The WebSphere Application Server will be run under root and will require access to the DB2 environment so that it can access the WebSphere Application Server administration database. This requires that the root account's environment .bashrc file be edited to add the following content at the end of the file:

```
# Setup DB2 environment for root user.
if [ -f /home/db2inst1/sqllib/db2profile ] ; then
. /home/db2inst1/sqllib/db2profile
fi

# Force DB2 to use JDBC 2.0.
if [ -f /home/<db2_instance_owner>/sqllib/java12/usejdbc2 ] ; then
. /home/<db2_instance_owner>/sqllib/java12/usejdbc2
fi
```

4.4.6 Configure and verify the DB2 client and server connectivity

To configure the DB2 Client, complete the following steps on the Web server:

1. Catalog the Database Server TCP/IP node as follows:

```
# su - db2inst1
$ db2 catalog tcpip node db2lx1 remote db2lx1 server 50000
```

Syntax:

```
> db2 catalog tcpip node <node_name> remote <db_server_hostname> server
<service/port#>
```

2. Catalog the WebSphere Application Server repository database as follows:

   ```
   $ db2 catalog db was40 at node db2lx1
   ```

   Syntax:

   ```
   > db2 catalog db <database_name> at node <node_name>
   ```

3. Verify the attach to the TCP/IP node as follows:

   ```
   $ db2 attach to db2lx1 user db2inst1 using <your_password>
   ```

   Syntax:

   ```
   > db2 attach to <node_name> user <db2inst_ID> using <db2inst_password>
   ```

4. Verify the connection to the database from the WebSphere Application Server (comlx1) as follows:

   ```
   $ db2 connect to was40 user db2inst1 using <your_password>
   ```

   Syntax:

   ```
   > db2 connect to <db_name> user <db2inst_ID> using <db2inst_password>
   ```

4.5 IBM HTTP Server and WebSphere plugin installation

In this scenario, the IBM HTTP Server V1.3.19.3 and WebSphere plugin V4.0.4 plus e-fixes are installed and configured on the commerce application server node. For detailed information, refer 3.4, “IBM HTTP Server and WebSphere plugin installation” on page 57.

4.6 WebSphere Application Server installation

In this scenario, the WebSphere Application Server V4.0.4 is installed and configured on the commerce application server node. For detailed information, refer 3.5, “WebSphere Application Server installation” on page 69.

4.7 WebSphere Commerce installation

In this scenario, the WebSphere Commerce V5.4 is installed on the commerce application server node. For detailed information, refer 3.6, “WebSphere Commerce installation” on page 83.

4.8 WebSphere Commerce Payments installation

In this scenario, the WebSphere Commerce Payments V3.1.3 is installed on the commerce application server node. For detailed information, refer 3.7, “WebSphere Commerce Payments installation” on page 84.

Note: Prior to installing WebSphere Payment Manager, as a prerequisite you will need to create and catalog the payman database. When referring to 3.7, “WebSphere Commerce Payments installation” on page 84, take note of the following slight modifications to the procedure to take into account the remote DB2 server node name db2lx1. Please refer to the following tips for instructions.

On the DB2 server:

1. Catalog the TCP/IP node.

   This step should already have been performed for the WebSphere Application Server configuration.

   ```
   $ db2 catalog tcpip node db2lx1 remote db2lx1 server db2cdb2inst1
   ```

2. Verify the attach to the TCP/IP node:

   ```
   $ db2 attach to db2lx1 user db2inst1 using db2inst1
   ```

3. Catalog the payman database.

   The payman database needs to be catalogued (for example, alias payman1) as part of this TCP/IP node:

   ```
   $ db2 catalog db payman as payman1 at node db2lx1
   ```

4. Verify the connection to the remote payman1 database from the DB2 client to the DB2 server.

   ```
   $ db2 connect to payman1 user db2inst1 using db2inst1
   ```

On the DB2 client:

1. Catalog the TCP/IP node.

   This step should already have been performed for the WebSphere Application Server configuration.

   ```
   $ db2 catalog tcpip node db2lx1 remote db2lx1 server 50000
   ```

   Where 50000 is the connection port in /etc/services of the DB2 server for the DB2 instance (it can also be a service name if added to the services file).

2. Verify the attach from the DB2 client to the DB2 server.

   ```
   $ db2 attach to db2lx1 user db2inst1 using <your_password>
   ```

3. Catalog the payman1 database on the DB2 client so that the WebSphere Payment Manager application installer can see the database on the database server.

   ```
   $ db2 catalog payman1 at node db2lx1
   ```

4. Verify the connection to the remote payman1 database from the DB2 client to the DB2 server.

```
$ db2 connect to payman1 user db2inst1 using <your_password>
```

4.9 WebSphere Commerce post-install configuration

In this scenario, the post-install configuration is performed on the commerce application server node. For detailed information, refer 3.8, “WebSphere Commerce post-install configuration” on page 89.

4.10 WebSphere Commerce instance creation

When creating a WebSphere Commerce instance in a remote database runtime environment, there are some minor differences to point out within the Configuration Manager Instance Creation Wizard. On the Database tab, select **Remote Database**. When this option is checked, additional fields on the Database tab will become accessible.

Database tab:

- Database administrator name: `db2inst1`

- Database administrator password: `<password>`
- Database name: `wc1db`
- Database type: `DB2`
- Database user name: `db2inst1`
- Database user password: `<password>`
- Select **Run Database Performance Wizard**
- Select **Set as active Database**
- Select **Use remote database**
- Database server hostname: `db2lx1`
- Database server port: `50000` (DB2 server connection port in services file)
- Dabatase node name: `db2lx1`

For detailed information, refer 3.9, “WebSphere Commerce instance creation” on page 100.

4.11 WebSphere Commerce post-instance config

For detailed information, refer 3.10, “WebSphere Commerce post-instance config” on page 105.

4.12 WebSphere Commerce Payments configuration

For detailed information, refer 3.11, “WebSphere Commerce Payments configuration” on page 110.

4.13 WebSphere Commerce runtime verification

For detailed information, refer 3.13, “WebSphere Commerce runtime verification” on page 113.

5

Implementing a 3-tier runtime with the addition of a remote Web server

The scenario highlighted in this chapter describes how to implement a 3-tier runtime environment by adding a remote Web server to an existing 2-tier remote database server runtime.

In this scenario, the IBM HTTP Server and WebSphere plugin are installed on node A. The WebSphere Application Server, WebSphere Commerce, WebSphere Commerce Payments, and DB2 Client are installed on node B. The DB2 Server is installed on node C.

The chapter is organized as follows:

- Planning for the 3-tier remote Web server runtime
- Red Hat Linux installation
- Install the remote IBM HTTP Server
- Configure the WebSphere plugin
- Configure Web server for WebSphere Commerce
- WebSphere HTTP transport SSL configuration

5.1 Planning for the 3-tier remote Web server runtime

This chapter provides detailed instructions for adding a remote IBM HTTP Server. The remote Web server is beneficial for security reasons. For more detailed information on security and remote Web servers, refer to the *WebSphere Commerce V5.4 Handbook*, SG24-6567.

There are two possibilities for adding a remote Web server:

- Add a remote Web server to a single-tier runtime environment to create a two-tier remote Web server runtime environment (Figure 5-1).
- Add a remote Web server to a two-tier remote database runtime environment to create a three-tier runtime environment (see Figure 5-2). This option is documented in this chapter.

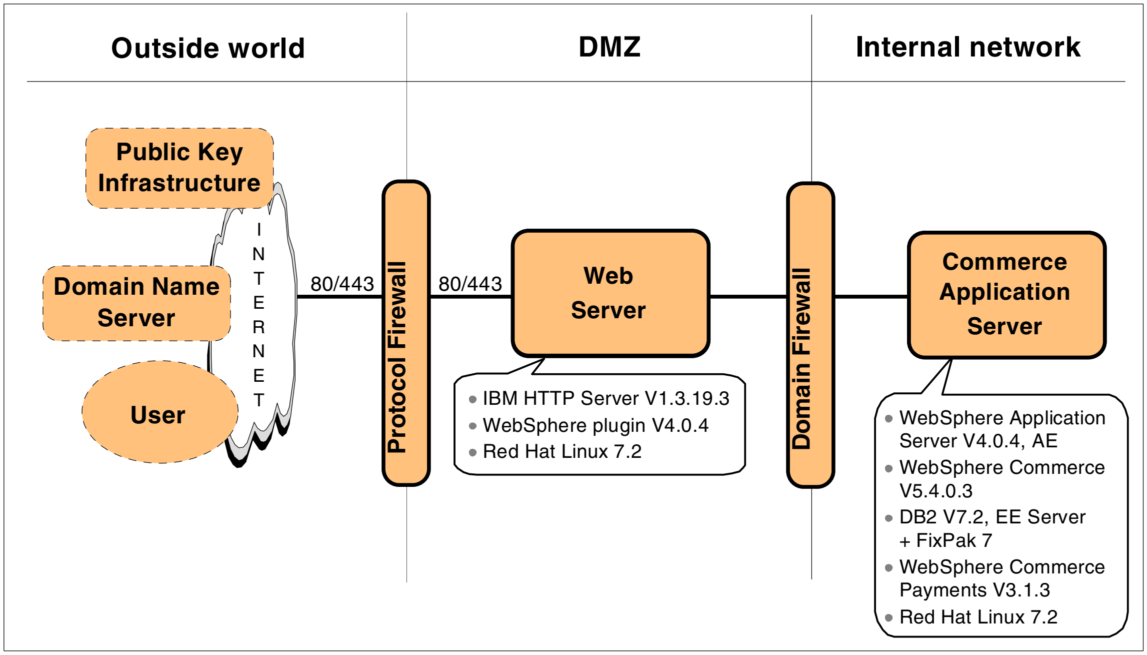

Outside world
DMZ
Internal network
Public Key Infrastructure
Domain Name Server
User
INTERNET
80/443
Protocol Firewall
80/443
Web Server
IBM HTTP Server V1.3.19.3
WebSphere plugin V4.0.4
Red Hat Linux 7.2
Domain Firewall
Commerce Application Server
WebSphere Application Server V4.0.4, AE
WebSphere Commerce V5.4.0.3
DB2 V7.2, EE Server + FixPak 7
WebSphere Commerce Payments V3.1.3
Red Hat Linux 7.2

Figure 5-1 Product mapping for adding a remote IBM HTTP Server

We provide detailed instructions for combining the procedures documented in the remote DB2 database server 2-tier and adding a remote IBM HTTP Server to create a 3-tier WebSphere Commerce runtime environment for Linux. The product mapping for the 3-tier runtime is depicted in Figure 5-2.

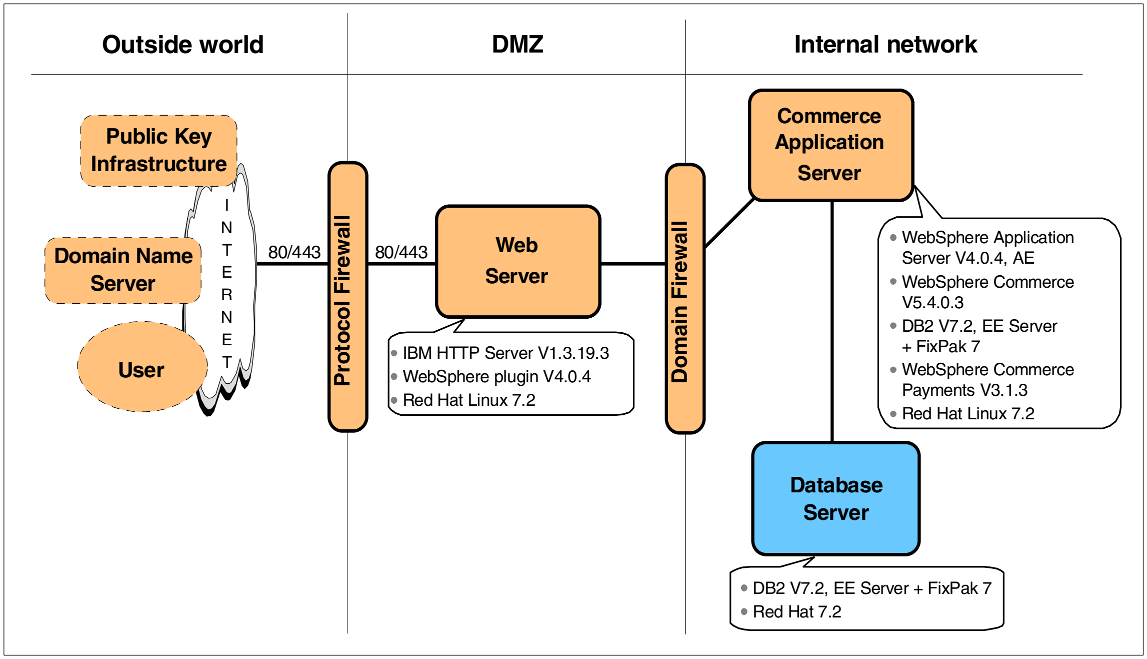

Outside world
DMZ
Internal network
Public Key Infrastructure
Domain Name Server
User
INTERNET
80/443
Protocol Firewall
80/443
Web Server
IBM HTTP Server V1.3.19.3
WebSphere plugin V4.0.4
Red Hat Linux 7.2
Domain Firewall
Commerce Application Server
WebSphere Application Server V4.0.4, AE
WebSphere Commerce V5.4.0.3
DB2 V7.2, EE Server + FixPak 7
WebSphere Commerce Payments V3.1.3
Red Hat Linux 7.2
Database Server
DB2 V7.2, EE Server + FixPak 7
Red Hat 7.2

Figure 5-2 Product mapping for a 3-tier WebSphere Commerce runtime environment for Linux

This chapter describes how to install, configure and verify a remote IBM HTTP Server for use with WebSphere Commerce. The chapter assumes that the user has already installed WebSphere Commerce in a single-tier or 2-tier remote DB2 configuration.

In this scenario, the IBM HTTP Server and the WebSphere plugin will be installed on a separate node, remote from WebSphere Application Server. The Web server should be configured to serve the static content, and not any WebSphere application code. The WebSphere plugin will forward any requests for dynamic content to the application server machine.

For more detailed information on many of the installation procedures, refer to the *Installation Guide, IBM WebSphere Commerce V5.4 Professional and Business Edition for Linux* product guide.

5.2 Red Hat Linux installation

On the remote Web server node, we installed Red Hat Linux 7.2. For detailed information, refer 3.2, “Red Hat Linux installation” on page 36.

5.3 Install the remote IBM HTTP Server

This section describes how to install and configure the remote IBM HTTP Server V1.3.19.3.

5.3.1 Install the IBM HTTP Server

For detailed information, refer to 3.4.1, “Install the IBM HTTP Server” on page 58.

5.3.2 Install the WebSphere plugin for IBM HTTP Server

For detailed information, refer to 3.4.2, “Install the WebSphere plugin for the IBM HTTP Server” on page 59.

5.3.3 Install WebSphere Application Server V4 FixPak 4

For detailed information, refer to 3.4.3, “Install the WebSphere Application Server V4 FixPak 4” on page 61.

5.3.4 Install WebSphere Application Server plugin e-fix

For detailed information, refer to 3.4.4, “Cumulative WebSphere plugin e-fix (PQ63508)” on page 61.

5.3.5 Configure the IBM HTTP Server

For detailed information, refer to 3.4.5, “Configure the IBM HTTP Server” on page 62.

5.3.6 Restart, stop and start the IBM HTTP Server

For detailed information, refer to 3.4.6, “Restart, stop and start IBM HTTP Server” on page 67.

5.3.7 Verify the IBM HTTP Server

For detailed information, refer to 3.4.7, “Verify the IBM HTTP Server” on page 67.

5.4 Configure the WebSphere plugin

This section describes how to configure the WebSphere Application Server for use with the remote IBM HTTP Server and includes the following topics:

- Start the WebSphere Application Server
- Add host aliases to virtual host
- Regenerate the Web server plugin
- Copy plugin-cfg.xml to the remote IBM HTTP Server

5.4.1 Start the WebSphere Application Server

To start the WebSphere Application Server, on the Commerce Application Server node, do the following:

1. Ensure that the DB2 UDB server is running and the was40 database is accessible.
2. Start the WebSphere Administrative Server.

   ```
   # su - wasuser
   $ cd /opt/WebSphere/AppServer/bin
   $ ./startupServer.sh &
   ```

3. View the contents of the tracefile as the WebSphere Administrative Server is initializing as follows:

   ```
   $ tail -f /opt/WebSphere/AppServer/logs/tracefile
   ```

   If the server starts successfully, you should see a message like the following:

   ```
   Server_adminServer open for e-business
   ```

5.4.2 Add host aliases to virtual host

To add host aliases to a virtual host on the WebSphere Application Server, do the following:

1. Start the WebSphere Application Server Administrative Console as follows:

   ```
   # su - wasuser
   $ cd /opt/WebSphere/AppServer/bin
   $ ./adminclient.sh
   ```

2. Add the following host aliases to the WebSphere Virtual Hosts.

   a. Under WebSphere Administrative Domain, select **Virtual Hosts**.

   b. Select the **General** tab.

   c. Add the following host aliases listed in to the default_host virtual host.

Table 5-1 Add host aliases to default_host virtual host

Host alias	Example
<web_server_hostname>:80	weblx1:80
<web_server_fully_qualified_hostname>:80	weblx1.itso.ral.ibm.com:80
<web_server_IP_address>:80	9.24.104.17:80
<web_server_hostname>:443	weblx1:443
<web_server_fully_qualified_hostname>:443	weblx1.itso.ral.ibm.com:443
<web_server_IP_address>:443	9.24.104.17:443

> **Note:** In this scenario, the hostname and IP address are those of the remote Web server.

> **Note:** The VH_<instance>_tools virtual host should be created automatically by WebSphere Commerce when creating an instance using the Configuration Manager.

d. When done adding virtual hosts, click **Apply**.

5.4.3 Remove and update host aliases from virtual hosts

In this scenario, we had the Web server installed on the same system as the application server. Now that we have added a remote Web server and have added host aliases for the remote Web server, we need to delete the old host aliases associated with the old Web server configured in scenario A.

To remove the host aliases from the virtual hosts associated with the old Web server, do the following:

1. On the Commerce Application Server, start the WebSphere Application Server Administrative Console.
2. Remove the following host aliases to the virtual host, where `<hostname>` is the host name of the scenario A Web server:

```
<hostname:80>
<hostname:443>
<fully_qualified_hostname:80>
<fully_qualified_hostname:443>
<IP_address:80>
<IP_address:443>
```

3. Update the following host aliases to the VH_<*instancename*>_tools:

```
<hostname:8000>
<fully_qualified_hostname:8000>
<IP_address:8000>
```

4. Restart the application servers that use the default_host and the VH_*instancename*_tools virtual hosts.

5.4.4 Regenerate the Web server plugin

After the remote Web server hostname has been added to the virtual host, we need to regenerate the Web server plugin. The plugin will later be copied to the remote Web server. Incoming requests to the remote Web server will be redirected to the application server based on the definitions within the plugin.

To regenerate the Web server plugin, do the following:

1. From the WebSphere Application Server Administrative Console, select and expand **WebSphere Administrative Domain -> Nodes -> <hostname>**.
2. With the host selected, right-click **Regen Webserver Plugin**.

> **Note:** When the Web server plugin is regenerated, the new virtual host aliases for the WebSphere Application Server are written to the plugin-cfg.xml file found in the /opt/WebSphere/AppServer/config directory. This file is loaded at startup or reload of the IBM HTTP Server with the WebSphere plugin-in.

3. Restart the Default Server application server from the WebSphere Application Server Administrative Console.

Stop the commerce node IBM HTTP Server

Now that the remote Web server is operational, we no longer need or want the IBM HTTP Server installed on the Commerce Application Server node.

1. Log on as root, and start a terminal session.
2. Stop the IBM HTTP Server by entering the following commands:

```
cd /opt/IBMHTTPServer/bin
./apachectl stop
```

5.4.5 Copy plugin-cfg.xml to the remote IBM HTTP Server

The newly regenerated plugin-cfg.xml contains the remote Web server hostname and IP address.

1. Copy /opt/WebSphere/AppServer/config/plugin-cfg.xml to the same directory on the remote IBM HTTP Server.
2. Restart the remote IBM HTTP Server and Default Server application server.

Verify the remote IBM HTTP Server and WebSphere

To verify the functionality of the remote IBM HTTP Server and WebSphere Application Server runtime, do the following:

1. Start the Default Server application server in the WebSphere Application Server Administrative Console.
2. After the Default Server has started, test the snoop servlet on both ports 80 and 443 from a Web browser client:

   ```
   http://<web_server_fully_qualified_hostname>/servlet/snoop
   https://<web_server_fully_qualified_hostname>/servlet/snoop
   ```

   Both URLs should return information from the servlet.

5.5 Configure Web server for WebSphere Commerce

Once the remote Web server and WebSphere Application Server have been configured and verified, now we need to configure the remote Web server for use with WebSphere Commerce administration tools and the store.

The high-level steps to configure the remote Web server for WebSphere Commerce, are as follows:

1. Copy static content to the remote IBM HTTP Server
2. Add entries to httpd.conf for WebSphere Commerce
3. Configure WebSphere Commerce Payments
4. Verify the WebSphere Commerce multi-tier runtime
5. Verify the WebSphere Commerce multi-tier runtime

5.5.1 Copy static content to the remote IBM HTTP Server

In the single-tier configuration, the WebSphere Commerce store and administration tools static content and Java code were already deployed on the same system. In this remote Web server configuration, we need to configure the runtime so that the static content for the store and administration tools are deployed on the remote Web server, which is more efficient at serving these types of Web assets.

Copy the static tools content to the remote Web server

The resources for your installed WebSphere Commerce instance are stored in an EAR subdirectory.

For example:

```
/opt/WebSphere/AppServer/installedApps/WC_Enterprise_App_<instance>.ear
```

The directory structure within the EAR file directory must be preserved. For simplicity, you should create new directories on the remote Web server machine to match the original paths on the Commerce Application Server node.

> **Note:** It is necessary to complete the first two of the following four steps each time you change the Web assets on your WebSphere Commerce server machine. Each time, remove all JSPs, META-INF and WEB-INF directories from the copied directory and its subdirectories.

To copy the static content from the Commerce Application Server node to the remote Web server node, do the following:

1. Create the following directories on the remote Web server for the static administration tools content, Web editor content (we do not recommend using this utility), and store static content on the remote Web server:

   a. Tools directory:

   ```
   # mkdir -p
   /opt/WebSphere/AppServer/installedApps/WC_Enterprise_App_<instance>.ear/
   wctools.war
   ```

   b. Web Editor directory:

   ```
   # mkdir -p
   /opt/WebSphere/AppServer/installedApps/WC_Enterprise_App_<instance>.ear/
   wcwebeditor.war
   ```

c. Stores directory:

```
# mkdir -p
/opt/WebSphere/AppServer/installedApps/WC_Enterprise_App_<instance>.ear/
wcstores.war
```

2. Copy the following directories, including subdirectories, from the Commerce Application Server node to the remote Web server node:

   a. Tools static content:

   ```
   /opt/WebSphere/AppServer/installedApps/WC_Enterprise_App_<instance>.ear/
   wctools.war
   ```

   b. Web Editor static content:

   ```
   /opt/WebSphere/AppServer/installedApps/WC_Enterprise_App_<instance>.ear/
   wcstores.war
   ```

   c. Stores directory:

   ```
   /opt/WebSphere/AppServer/installedApps/WC_Enterprise_App_<instance>.ear/
   wcwebeditor.war
   ```

Note: We created a tar file containing each of the above listed <name>.war files (for example, wctools.war, wcwebeditor.war).

For example: wctools.war

- Create a wctools.war.tar file and put it in the /tmp directory:

  ```
  # tar -cvf /tmp/wctools.war.tar
  /opt/WebSphere/AppServer/installedApps/WC_Enterprise_App_<instance>.ea
  r/wctools.war
  ```

- Ensure the equivalent directory exists on the remote Web server.
- Copy the tar file to the remote Web server.
- Untar the tar file on the remote Web server.

  ```
  # cd
  /opt/WebSphere/AppServer/installedApps/WC_Enterprise_App_<instance>.ea
  r
  # tar -xvf wctools.war.tar
  ```

3. Optionally, copy the WebSphere Application Server contents of the following directories including subdirectories, from the Commerce Application Server node to the remote Web server node:

```
/opt/WebSphere/CommerceServer/web/doc
/opt/WebSphere/AppServer/web
/opt/WebSphere/AppServer/WSsamples
```

4. Remove all the JSPs located on the Web server machine in the /opt/WebSphere/AppServer directory and its subdirectories. Remove META-INF and WEB-INF subdirectories.

Note: We suggest using a command and/or script to remove the files and directories. Be careful with the **`remove`** command. Substitute the WebSphere Commerce instance with your own instance name.

For example, create a script like the following:

```
#!/bin/ksh
find /opt/WebSphere/AppServer/installedApps/WC_Enterprise_Apps_wc1.ear \(
-name META-INF -o -name WEB-INF -o -name '*.jsp' \) \-exec rm -r {} \;
```

Copy store static content to the remote IBM HTTP Server

This section describes how to copy the static store content to the remote Web server. Replace `<instance>` with your instance name.

1. Ensure the following directory for the static store content exists on the remote Web server (should have been created there):

   ```
   # mkdir -p
   /opt/WebSphere/AppServer/installedApps/WC_Enterprise_App_<instance>.ear/wcs
   tores.war
   ```

2. Copy the following directories, including subdirectories, from the Commerce Application Server node to the remote Web server node:

   ```
   /opt/WebSphere/AppServer/installedApps/WC_Enterprise_App_<instance>.ear/wcs
   tores.war
   ```

   Note: We created a tar file containing each of the above listed <name>.war files; for example, wcstores.war.

   - Create a wcstores.war.tar file and put it in the /opt directory:

     ```
     # tar -cvf /tmp/wcstores.war.tar
     /opt/WebSphere/AppServer/installedApps/WC_Enterprise_App_<instance>.ea
     r/wcstores.war
     ```

   - Ensure the equivalent directory exists on the remote Web server.
   - Copy the tar file to the remote Web server.
   - Untar the tar file on the remote Web server.

     ```
     # cd /opt/WebSphere/AppServer/installedApps/WC_Enterprise_App_wc1.ear
     # tar -xvf wcstores.war.tar
     ```

3. Remove all the JSPs located on the Web server machine in the /opt/WebSphere/AppServer directory and its subdirectories. Remove META-INF and WEB-INF subdirectories. Refer to the note above to automate this step.

Note: We suggest using a command and/or script to remove the files and directories. Be careful with the **remove** command. Substitute the instance with your own instance name.

```
# su - root
# find
/opt/WebSphere/AppServer/installedApps/WC_Enterprise_App_<instance>.ea
r \( -name META-INF -o -name WEB-INF -o -name *.jsp \) \-exec rm -r {}
\;
```

5.5.2 Add entries to httpd.conf for WebSphere Commerce

When a WebSphere Commerce instance is created, Configuration Manager modifies the httpd.conf with aliases and directives needed for WebSphere Commerce. In the single-tier configuration, the Web server was on the same system as WebSphere Commerce and this was not a problem.

In a multi-tier environment with a remote Web server, we need to manually add these entries to the httpd.conf for WebSphere Commerce to the remote IBM HTTP Server httpd.conf file. This can be accomplished by copying the httpd.conf from the single-tier Web server set up in the previous scenario or by manually adding the entries (or by cutting and pasting).

Add virtual hosts, aliases, and directives to httpd.conf for WebSphere Commerce.

1. Stop the IBM HTTP Server service by entering the following command:

   ```
   /opt/IBMHTTPServer/bin/apachectl stop
   ```

2. To save time, copy the httpd.conf we already have working on the httpd.conf with the WebSphere Commerce entries from the single-tier setup. Search for and replace the host name and IP address for the new Web server.
3. Restart the IBM HTTP Server.
4. Restart the WebSphere Commerce <instance> application server using the WebSphere Application Server Administration Console.
5. Verify that you are still able to access the IBM HTTP Server via SSL:

   ```
   https://<webserver_hostname>
   ```

5.5.3 Configure WebSphere Commerce Payments

This section describes the necessary configuration steps to use WebSphere Commerce Payments in a multi-tier configuration (remote Web server), where WebSphere Commerce Payments is installed on the same system as WebSphere Commerce.

The high-level configuration steps are as follows:

1. Change the host name in the WCSRealm.properties file
2. Change the host name for the application server
3. Start WebSphere Payment Manager

Change the host name in the WCSRealm.properties file

In the single-tier scenario, the Web server was installed on the same system as WebSphere Commerce and WebSphere Payment Manager. The WCSRealm.properties file contains the host name of the Web server.

Now that we have a remote Web server, the host name must be updated in the WCSRealm.properties file for WebSphere Payment Manager.

To update the host name in the WCSRealm.properties file, do the following:

1. Stop the WebSphere Payment Manager server and application server using the WebSphere Application Server Administration Console.
2. Log on as root and open a terminal session.
3. Change to the following Payment Manager home directory:

   ```
   cd /opt/PaymentManager
   ```

4. Open the WCSRealm.properties file with a text editor (such as vi).
5. Change the value for the WCSHostName to the host name of the remote Web Server.

Change the host name for the application server

To change the host name for the WebSphere Payment Manager application server, do the following:

1. Start the WebSphere Application Server Administrative Console.
2. Select the **WebSphere Payment Manager** application server.
3. Select the **JVM Settings** tab.
4. Change the value of the wpmui.PaymentServerHostname to the fully qualified host name of the remote Web server.
5. Start the WebSphere Payment Manager application server from the WebSphere Application Server Administrative Console.

Start WebSphere Payment Manager

In order to make WebSphere Payment Manager (IBMPayServer) use the remote Web server host name instead of the local host name on which it is installed, we need to enter parameters when starting the IBMPayServer.

To start the IBMPayServer with the remote Web server host name, do the following:

1. Start a terminal session and change to the <payment_manager_home> directory:

   ```
   cd /opt/PaymentManager
   ```

2. Enter the following command to start WebSphere Payment Manager:

   ```
   #./IBMPayServer -pmhost <remote_web_server_hostname>
   ```

3. When you are prompted for a password, enter the WebSphere Payment Manager database password.

5.5.4 Verify the WebSphere Commerce multi-tier runtime

In order to verify that the WebSphere Commerce multi-tier implementation is functioning correctly, perform the following steps.

Verify the WebSphere Commerce administration tools

Check the following URLs from a Web browser:

```
– https://<webserver_hostname>:8000/adminconsole
– https://<webserver_hostname>:8000/storeservices
– https://<webserver_hostname>:8000/accelerator
```

Verify the WebSphere Commerce store

Check the following URL in a Web browser:

```
http://<remmote_web_server_hostname>/webapp/wcs/stores/servlet/StoreCatalog
Display?storeId=10001&catalogId=10001&langId=-1
```

Tip: Alternatively, you can launch the store URL from the Storeservices menu. Click **Publish Summary** from the right panel and then click **Launch Store**.

5.6 WebSphere HTTP transport SSL configuration

At this point, the communication transport between the remote IBM HTTP Server and the Commerce Application Server is not secure. In this scenario, we use SSL to encrypt the communication between the IBM HTTP Server WebSphere plug-in and the HTTP transport of the Web container on the WebSphere Application Server. SSL is configured to require the HTTP transport (server) to authenticate itself to the plug-in (client).

In order to create a secure connection between the two servers, we need to configure SSL in one of two possible configurations:

- Client authentication enabled
- Client authentication disabled

In this section, we describe the steps necessary to configure SSL with client authentication disabled. For more information on this topic, refer to the *WebSphere V4.0 Handbook*, SG24-6176.

The high-level steps to configure the WebSphere HTTP transport to use SSL with client authentication disabled are as follows:

1. Create a WebSphere Application Server key database
2. Create a certificate for the WebSphere Application Server
3. Configure the WebSphere transport
4. Create a IBM HTTP Server plug-in key database
5. Extract the WebSphere certificate
6. Import WebSphere certificate as CA on IBM HTTP Server
7. Regenerate plugin-cfg.xml
8. Restart IBM HTTP Server

5.6.1 Create a WebSphere Application Server key database

To create a WebSphere Application Server key database file using the JKS format, do the following:

1. Start the IKeyMan utility provided with the WebSphere Application Server.

   ```
   # cd /opt/WebSphere/AppServer/bin
   # ./ikeyman.sh
   ```

2. From the menu bar, click **Key Database File -> New**.

3. When the New window appears, enter the following and then click **OK**:

   – Key database type: select **JKS**
   – File name: `was_key.jks`
   – Location: `/opt/WebSphere/AppServer/etc`

4. When the Password Prompt window appears, enter the following and then click **OK**:
   - Password: <your_password> (used to protect the key store file)
   - Select **Set expiration time?** and enter the number of days before the password should expire. If no expiration is required, do not select this setting.

Note: Although it is not required in a development environment, it is strongly recommended that all keystores used in a production environment set an expiration period. Also, remember this password, because it will be needed for the WebSphere transport settings window in a later step.

5.6.2 Create a certificate for the WebSphere Application Server

In this scenario, we created a self-signed certificate for the WebSphere Application Server, as follows:

1. Start the IKeyMan utility provided with the WebSphere Application Server.
2. From the menu bar, click **Key Database File -> Open**.
3. Specify the `/opt/WebSphere/AppServer/etc/was_key.jks` file.
4. While selecting the key database file you just opened/created, create a new Self-Signed Certificate by clicking **Create -> Create New Self-Signed Certificate** from the menu bar.

Note: If you are enabling SSL for a production environment, select **New Certificate Request** instead. It is recommended that self-signed digital certificates not be used in production.

5. When the Create New Self-Signed Certificate window appears, enter the following and then click **OK**:
   - Key Label: `<user_defined_label>` (for example, `was_ssl`)
   - Common Name: `<was_fully_qualified_hostname>` (for example, `m23vnx88.itso.ral.ibm.com`)
   - Organization: `<your_organization>` (for example, `ibm`)
6. When done, close the IBM Key Management (IKeyMan) utility.

5.6.3 Configure the WebSphere transport

To create a new secure SSL enabled transport for the WebSphere Commerce application server, do the following:

1. Start the WebSphere Application Server Administrative Console.
2. Select the **WebSphere Commerce - <instance>** application server and right-click **Stop**.
3. With the WebSphere Commerce - <instance> application server selected, select the **Services** tab.
4. Select the **Web Container Service** from the Services list, and then click **Edit Properties**.
5. Select the **Transport** tab.
6. From the Transport tab, click **Add** to add a new HTTP transport.
7. From the General tab, enter the following properties and click **OK**:
   - Transport host: *
   - Transport port: `9083` (must not be in use on the WebSphere Application Server system)
   - Enable SSL: select the **Enable SSL** checkbox
   - Use global SSL configuration: do not select to disable (we will be using our own key store and settings)
   - Key file name: `/opt/WebSphere/AppServer/etc/was_key.jks`
   - Key file password: `<your_keystore_password>` (password used to protect key store)
   - Confirm password: `<your_keystore_password>`
   - Key file format: `JKS`
   - Enable client authentication: do not select (disable)
   - Security level: `high`
8. In the Web Container Service window, there is a new (9083) transport. Click **OK**.
9. In the Properties window of Administrative Console, click **Apply**.
10. Right-click the **WebSphere Commerce - <instance>** application server and select **Start**.

    The status of the new HTTPS transport is indicated by an entry in the Administrative Console Event Messages window.

5.6.4 Create a IBM HTTP Server plug-in key database

To create a IBM HTTP Server plug-in key database on the remote Web server, refer to “Create a key database for the IBM HTTP Server” on page 65. In our scenario, we used the existing key database created when configuring the remote IBM HTTP Server for SSL.

5.6.5 Extract the WebSphere certificate

On the Commerce Application Server, extract the certificate as follows:

1. Start the IKeyMan utility provided with the WebSphere Application Server.
2. From the menu bar, click **Key Database File -> Open**.
3. Specify the `/opt/WebSphere/AppServer/etc/was_key.jks` file.
4. Enter the key database password and click **OK**.

   You should see the certificate listed (in our example, was_ssl).
5. Select the certificate (**was_ssl**) and click **Extract Certificate** to export the certificate as a Base64-encoded ASCII file.
6. When the Extract Certificate window appears, enter the following and then click **OK**:
   - Data type: select **Base64-encoded ASCII data**
   - Certificate file name: `was_cert.arm`
   - Location: `/opt/WebSphere/AppServer/etc`
7. Close the IBM Key Management utility.

5.6.6 Import WebSphere certificate as CA on IBM HTTP Server

To import the WebSphere Application Server certificate as a trusted CA on the remote IBM HTTP Server, do the following:

1. Copy the WebSphere certificate file (`/opt/WebSphere/AppServer/`etc/was_cert.arm) to the same directory (`/opt/WebSphere/AppServer/`etc) on the remote IBM HTTP Server.
2. Start the IBM Key Management utility on the remote IBM HTTP Server.
3. From the menu bar, select **Key Database File -> Open**.
4. Specify the path to the IBM HTTP Server key database:

   ```
   /opt/IBMHTTP/ssl/http_key.kdb
   ```
5. Enter the key database password and click **OK**.

   You should see the IBM HTTP Server certificate listed (for example, http_ssl).

6. Under the Key database content pull-down, select **Signer Certificates**, and then click **Add**.
7. When the Add CA's Certificate from a File window appears, enter the following and then click **OK**:
   - Data type: select **Base64-encoded ASCII data**
   - Certificate file name: `was_cert.arm` (WebSphere Application Server exported certificate copied to the remote IBM HTTP Server)
   - Location: `/opt/WebSphere/AppServer/`etc (the path must be the same on the remote Web server and the application server)
8. When prompted, enter the label for the certificate was_cert.arm (for example, was_ssl) and then click **OK**.
9. The WebSphere Application Server CA certificate (was_ssl) should now be listed in the remote IBM HTTP Server trusted CA's list.
10. Close the IBM Key Management utility.

> **Note:** Make sure that the dates on the two machines match before trying to add the certificate, so that its validity period will be active when you add it. If they do not match and the date on the application server machine is ahead of the one on the Web server machine, then you might need to delete the certificate and recreate it on the application server machine after changing the dates, and then try to add it again.

Regenerate plugin-cfg.xml

After making all the necessary configuration changes to configure the WebSphere transport for SSL, we need to regenerate the plug-in.

Copy plugin-cfg.xml to the remote IBM HTTP Server

Copy the plugin-cfg.xml file from the commerce application server node to the same directory of the remote IBM HTTP Server node (/opt/WebSphere/AppServer/config).

Restart IBM HTTP Server

Restart the IBM HTTP Server service on the remote Web server.

5.6.7 Verify the configuration

Before proceeding to the next section, verify that the three-tier configuration is still working properly.

A

Linux tips

This appendix provides information on the Red Hat Linux installation, common Linux command and tasks, and where to find more information.

The appendix is organized into the following sections:

- Red Hat Linux 7.2 installation
- Common Linux commands and tasks
- Where to find more information

A.1 Red Hat Linux 7.2 installation

The objective of including an installation section for Red Hat Linux 7.2 is to show which options we selected during a typical installation for the scenarios documented in this Redpaper.

For more detailed information on the installation of Red Hat Linux 7.2, refer to the *Red Hat Linux Installation Guide* found at: http://www.redhat.com.

To install Red Hat Linux 7.2, complete the following steps:

1. Insert the Red Hat Linux 7.2 CD 1, and boot from the CD to start the installation.

   > **Note:** Configuring the system to boot from CD typically in the firmware configuration. On IBM systems, this can be accomplished by setting the CD ROM as the first bootable device in the firmware.

2. When prompted press Enter, to start the GUI installer.
3. When the Language Selection window appears, we selected **English**, and then click **Next**.
4. When the Keyboard Configuration window appears, we selected the following and then clicked **Next**:
   - Model: **Generic 105-key (Intl) PC**
   - Layout: **U.S. English**
   - Dead Keys: **Enable dead keys**
5. When the Mouse Configuration window appears, we selected the following and then clicked **Next**:
   - Model: **Generic 2 Button Mouse (PS/2)**

     This may be different for you system (for example, 3 button mouse)
6. When the Welcome to Red Hat Linux window appears, review the information and then click **Next**.
7. When the Install Options window appears, we selected **Custom**, and then clicked **Next**.
8. When the Choosing Your Partitioning Strategy window appears, we selected **Manually partition with Disk Druid** and then click **Next**.
9. When the Partitions window appears, create the necessary partitions and then click **Next**.

   When configuring partitions there are a couple of options to consider:

   - One large root file system (development or test system)

When configuring a development or test system, you may not be concerned about having unique files systems. In this case, create a swap file system type approximately 1-2 times the amount of RAM, and the remainder of the disk to be allocated to / (root) using the EXT file system type.

- Individual file systems (production)

  On a production system, it may be necessary and wise to create individual file systems and allocate a specific amount of storage to that file system using the EXT file system type.. For example, if you create a separate file system for your database with the necessary disk space, you can prevent a problem of file system being filled up by other applications. In this case only DB2 would have access to write data to the file system.

  Example file system mount points

  ```
  /
  /usr/local
  /home
  /var
  swap
  ```

  > **Note:** For details on the files system (partitions) we created for the WebSphere Commerce installation, refer to 3.2.3, “File systems” on page 37.

10. When the Boot Loader Installation window appears, we selected the following and then click **Next**:
    - Boot Loader Configuration: select **Use GRUB as the boot loader**
    - Install Boot Loader record on: select **/dev/sda Master Boot Record (MBR)**
    - Partition: check **Default boot image**
11. When the GRUB Password window appears, we accepted the default (no password) and click **Next**.
12. When the Network Configuration window appears, we entered the following and then clicked **Next**:
    - Configure using DHCP: uncheck (we used a static IP address)
    - Activate on boot: check (default)
    - IP Address: `9.24.105.112`
    - Netmask: `255.255.254.0`
    - Network: `9.24.104.0`
    - Broadcast: `9.24.105.255`
    - Hostname: `comlx1`
    - Gateway: `9.24.104.1`

- Primary DNS: `9.24.106.15`
- Secondary DNS: na
- Ternary DNS: na

13. When the Firewall Configuration window appears, we selected **No Firewall** and then clicked **Next**.
14. When the Language Support Selection window appears, we checked **English (USA)** and then click **Next**.
15. When the Time Zone Selection window appears, we selected **America/New_York** and then clicked **Next**.
16. When the Account Configuration window appears, we entered the following and then clicked **Next**:
    - Root Password: `<password>`
    - Confirm: `<password>`
    - Under Additional Accounts, click **Add** to add users. We added the user `admin`.
17. When the Authentication Configuration window appears, we accepted the defaults (Enable MD5 passwords, Enable shadow passwords) and click **Next**.
18. When the Selecting Package Groups window appears, we selected the following packages and then clicked **Next**:
    - Check **Printing Support** (default)
    - Check **Classic X Window System** (default)
    - Check **X Window System** (default)
    - Uncheck **Laptop Support** (default)
    - Check **GNOME** (default)
    - Check **KDE** (optional)
    - Uncheck **Sound and Multimedia Support**
    - Check **Network Support** (default)
    - Uncheck **Dialup Support** ‘
    - Check **Messaging and Web Tools** (default)
    - Uncheck **Graphics and Image Manipulation** (default)
    - Uncheck **New Server** (default)
    - Uncheck **NFS File Server** (default)
    - Uncheck **Windows File Server** (default)
    - Uncheck **Anonymous FTP Server** (default)
    - Uncheck **SQL Database Server** (default)
    - Uncheck **Web Server** (default)
    - Uncheck **Router/Firewall** (default)
    - Uncheck **DNS Name Server** (default)
    - Uncheck **Authoring and Publishing** (default)
    - Uncheck **Emacs** (default)
    - Uncheck **Utilities** (default)

- Uncheck **Legacy Application Support** (default)
- Uncheck **Software Development** (default)
- Uncheck **Kernal Development** (default)
- Uncheck **Windows Compatibility/Interoperability** (default)
- Uncheck **Games and Entertainment** (default)
- Uncheck **Everything** (default)

19. When the Video Configuration window appears, select your video hardware and then click **Next**.

20. When the About to Install window appears, review the caution about to install files message, and then click **Next** to begin installing packages.

21. When the Installing Packages window appears, you will see progress indicators for the file system formatting and package installation.

    During the installation you may be prompted to insert Red Hat Linux 7.2 CD2 depending on the options that you have selected.

22. When the Boot Disk Creation window appears, insert a blank formatted diskette and then click **Next**.

    > **Note:** Although we do recommend that you create a boot disk for recovery purposes, as an alternative you can check skip boot disk creation.

23. When the Monitor Selection window appears, select your monitor type and clcik **Next**.

24. When the Custom X Configuration window appears, we selected the following and then clicked **Next**:
    - Color Depth: select **High Color (16 Bit)**
    - Screen Resolution: select **1024x768**
    - Default desktop: select **GNOME**
    - Login type: select **Graphical**

25. You should now see a message `Congratulations your Red Hat Linux installation is now complete`. Click **Exit** to restart your system.

The Red Hat Linux 7.2 installation is now complete. We strongly recommend that you verify your network configuration before proceeding to install WebSphere Commerce and supporting software.

A.2 Network configuration

This section briefly describes how to verify that the Linux systems networking is working properly.

- Verify /etc/hosts file.

  If you do not have a host entry for the hostname of your node add it. For example, our /etc/hosts file contains the following:

  ```
  127.0.0.1 localhost.localdomain localhost
  9.24.105.112 comlx1.itso.ral.ibm.com comlx1
  ```

- Verify the /etc/resolv.conf file.

  This file defines the Domain Name Server (DNS), domain, and the domain search order. For example, out /etc/resolv.conf file contains the following:

  ```
  nameserver 9.24.106.15
  domain itso.ral.ibm.com
  search itso.ral.ibm.com
  ```

- Verify the network configuration using the `ifconfig` command.

  The `ifconfig` command returns

  ```
  # ifconfig
  ```

- Verify route

  ```
  # route -n
  ```

- Verfiy that you can access other systems and your system by name (resolves name with DNS or hosts file).

A.3 Common Linux commands and tasks

This section highlights common Linux commands and tasks.

Shutdown

Shutdown now without file system check, enter the following:

```
shutdown -f now
```

Shutdown and reboot without file system check, enter the following:

```
shutdown -f -r now
```

Mount a CD ROM

```
mount -r /dev/<device> /mnt/cdrom
```

or

```
mount /mnt/cdrom
```

Unmount a CD ROM

```
unmount /mnt/cdrom
```

Modify the video settings

```
Xconfigurator
Xconfiguration --help (information on command options)
```

Add users

```
adduser <username>
```

Add groups

```
groupadd <groupname>
```

Change a users password

```
passwd <user id>
```

Create new file system

```
mkfs /dev/sda2 2000000
```

Refer to the following URL for detail steps:

```
http://www.redhat.com/docs/manuals/linux/RHL-8.0-Manual/custom-guide/s1-filesystem-ext3-create.html
```

Monitor disk space and file system usage

```
df
du
```

Modify network configuration

```
ifconfig
route -n
nslookup
telnet localhost <port#>
/etc/xinetd.d (servers)
/etc/rc.d/init.d/xinetd restart (restart all services n xinetd.d directory)
```

Directory listing

```
ls -l (long form list)
ls -al (hidden files)
ls (short form list)
```

Show processes running

```
ps -axf
ps -axf -width=240
ps -ef
```

Find a file

```
locate <file name>
find <target directory> -name <file name> -print
```

Network restart commands: “ifdown”, “ifup”

```
ifdown eth0 (bring ethernet interface down)
ifup eth0 (bring ethernet interface up)
```

Utility to check, add or delete a service in /etc/init.d

```
chkconfig
```

gzip command

```
gzip -d -c <filename>.gz | tar -xvf -
```

FTP server configuration

To configure an FTP server on Red Hat Linux, do the following:

1. Install the `wu-ftpd-2.6.1-18.i386.rpm` found on Red Hat Linux CD 1.

   ```
   # mount -r /dev/cdrom /mnt/cdrom
   # rpm -U --nodeps wu-ftpd-2.6.1-18.i386.rpm
   ```

2. Modify the `/etc/xinetd.d/wu-ftpd` file by changing the disable value to no.
3. Restart the TCP/IP services.

   ```
   # /etc/rc.d/init.d/xinetd restart
   ```

Note: By default, the root user ID can not directly ftp to the FTP server.

A.4 Where to find more information

There are many sources available on the Web for Linux. We recommend the following URL as a starting place, found on the Red Hat Web site:

http://www.redhat.com/docs/manuals/

B

Automating server startup

This appendix describes how to automate the startup servers or daemons upon system restart, for servers used within the WebSphere Commerce runtime environment on the Linux platform.

This appendix includes the following topics:

- DB2 Server auto start
- IBM HTTP Server auto start
- WebSphere Application Server and WebSphere Commerce auto start
- WebSphere Commerce Payments auto start
- WebSphere Commerce Payments auto start

Auto start overview

The autostart scripts should be in the /etc/rc.d/init.d directory and they have to be executeable. Once they are there, you can start them issuing the following type of command: service service_name [start | stop] for example: service websphere start

In order to run the scripts during startup you have to make symbolic links. If you are running on run level 5 (with X11 we were running on this) the links should point to the files in the /etc/rc.d/rc5.d directory

For example:

ln -s /etc/rc.d/init.d/websphere /etc/rc.d/rc5.d/S23websphere

ln -s /etc/rc.d/init.d/websphere /etc/rc.d/rc5.d/K23websphere

In the links S stands for Start, K stands for Kill, these scripts start and kill the daemons during startup and shutdown. The numbers are determining the startup sequence (lower numbers start first).

DB2 Server auto start

Add the following entry to the /etc/inittab file after the # System initialization entry.

```
db:2345:once:/etc/rc.db2 > /dev/console 2>&1
```

Note: This entry is added at the end of the /etc/inittab when DB2 is installed. Verify the /etc/inittab to check if the entry already exist. If not, add the above entry.

The DB2 services need to be started before WebSphere Application Server, Commerce and Payment Applications. Make sure the DB2 entry is before these WebSphere applications autostart executions.

IBM HTTP Server auto start

In a single tier environment where the IBM HTTP Server and the Application Server are on the same machine, the IBM HTTP Server should be started before the WebSphere Application Server. To configure the IBM HTTP Server auto start script, you need to do the following:

- Verify that there is a a shell script program named `ibmhttpd` in the /etc/rc.d/init.d directory. The program is installed when you install IBM HTTP Server.
- create two symbolic link files in /etc/rc.d/rd5.d directory by issuing the following commands:

```
ln -s /etc/rc.d/init.d/ibmhttpd /etc/rc.d/rc5.d/S85ibmhttpd
ln -s /etc/rc.d/init.d/ibmhttpd /etc/rc.d/rc5.d/K20ibmhttpd
```

The system is now configured to run IBM HTTP Server automatically when the system starts.

WebSphere Application Server and WebSphere Commerce auto start

The following tasks provide you with guidelines on how to configure an auto start environment for WebSphere Application Server, WebSphere Commerce and Payment Application Servers.

- create a shell script program named `websphere` in the /etc/rc.d/init.d directory. An example of the script is provided as Example 5-1

Example 5-1

```
#!/usr/bin/ksh
# Neil Buckingham 24/7/00
#
# Can take a long long time so run last from init

USAGE="Usage:  $0 <start | stop>";
if [ $# -lt 1 ]; then
   echo $USAGE;
   exit 1;
fi

unset JAVA_HOME
unset JDKDIR
umask 002

action=$1;
case $action in
  start)
    if [ -d /opt/WebSphere/AppServer ]
    then

        echo "Starting: WebSphere Application Server (2 minutes): \c"
        cd /opt/WebSphere/AppServer/logs
```

```
      cp tracefile tracefile.-1
      >tracefile
      chmod 644 tracefile
      cd /opt/WebSphere/AppServer/bin
      ./startupServer.sh >../logs/init.log 2>&1 &
      COUNT="1"
      EXIT="OK"
      until grep 'WebSphere Administration server open for e-business'
../logs/tracefile >/dev/null 2>&1
      do
         if [ $COUNT -gt 600 ]
         then
            EXIT="FAILED"
            break
         fi
         COUNT=`expr $COUNT + 1`
         sleep 2
      done

      ./wscp.sh -c "ApplicationServer start
/Node:comlx1/ApplicationServer:wc1/"
      ./wscp.sh -c "ApplicationServer start
/Node:comlx1/ApplicationServer:payment/"

      echo "$EXIT"
   else
      echo "Could not start WebSphere Application Server"
   fi
  ;;
  stop)
   if [ -d /opt/WebSphere/AppServer ]
   then
      echo "Stopping: WebSphere Application Server (1 minute)"
      cd /opt/WebSphere/AppServer/bin
      echo '<?xml version="1.0"?>
      <!DOCTYPE websphere-sa-config SYSTEM
"$XMLConfigDTDLocation$$dsep$xmlconfig.dtd" >
      <websphere-sa-config>
      <node name="comlx1" action="stop">
      </node>
      </websphere-sa-config>' > /tmp/stopnode.xml
      ./XMLConfig.sh -adminNodeName comlx1 -import /tmp/stopnode.xml
>>../logs/init.log 2>&1
      sleep 15
   fi
  ;;
  *)
   echo $USAGE;
   exit 1;
  ;;
esac
exit 0;
```

- create two symbolic link files in /etc/rc.d/rd5.d directory by issuing the following commands:

```
ln -s /etc/rc.d/init.d/websphere /etc/rc.d/rc5.d/S99websphere
ln -s /etc/rc.d/init.d/websphere /etc/rc.d/rc5.d/K20websphere
```

The system is now configured to run WebSphere Applications (admin server, commerce server and payment server) automatically when the system starts.

WebSphere Commerce Payments auto start

If you need Payment Manager to be startable when there is no user present toenter the password, you need to create a password file. The default password file name is `.payment` and the file should reside in the same directory as the Payment Manager installation directory. The file should contain a single line as follows:

```
DBPassword=db2inst1
```

To start Payment Manager using this password file, open a command window and change directory to the Payment Manager installation directory. Then enter thefollowing command:

```
IBMPayServer -file
```

Instead of prompting you for the password, Payment Manager reads the password directly from the `.payment` file.

Alternatively, you can create a payment auto start shell script file, and follow the same configuration procedures as the WebSphere Applications auto start . A sample of payment auto shell script is provided as Example 5-2

Example 5-2

```
#!/usr/bin/ksh
# Payment Manager

action=$1;
case $action in
start)
   echo "Starting WebSphere Payment Manager: \c"
   /opt/PaymentManager/IBMPayServer db2inst1 > /opt/PaymentManager/payment.log
2>&1 &
```

```
   COUNT="1"
   EXIT="OK"

   until grep "IBM WebSphere Payment Manager has started successfully."
/opt/PaymentManager/payment.log >/dev/null 2>&1
   do
      if [ $COUNT -gt 10 ]
      then
         EXIT="FAILED"
         break
      fi
      COUNT=`expr $COUNT + 1`
      sleep 2
   done
   echo "$EXIT"

;;
stop)
   echo "Stopping WebSphere Payment Manager:"

   /opt/PaymentManager/StopIBMPayServer >/dev/null 2>&1
;;
esac
exit 0;
```

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this Redpaper.

IBM Redbooks

For information on ordering these publications, see “How to get IBM Redbooks” on page 168.

- *WebSphere Commerce V5.4 Handbook*, SG24-6567
- *WebSphere Commerce V5.4 Developers Handbook*, SG24-6190
- *e-commerce Patterns for Building B2C Web Sites Using IBM WebSphere Commerce Suite V5.1*, SG24-6180
- *B2B e-Commerce Composite Pattern Using IBM WebSphere Commerce Suite V5.1, Business Edition*, SG24-6181
- *e-commerce Payment Solutions Implementation and Integration Using IBM WebSphere Payment Manager*, SG24-6177
- *Mobile Commerce Solutions Guide using WebSphere Commerce Suite V5.1*, SG24-6171
- *WCS V5.1 Performance Tuning*, SG24-6258

Other resources

These publications are also relevant as further information sources:

- *Installation Guide, IBM WebSphere Commerce V5.4 Professional and Business Edition for Linux*
- *Additional Software Guide, IBM WebSphere Commerce V5.4 Professional and Business Edition for Linux*
- *Fundamentals Guide, IBM WebSphere Commerce V5.4*
- *WebSphere Commerce V5.4 Online Help*
- *Installation Guide, IBM WebSphere Payment Manager for Multiplatforms*
- Jonathan Adams, et al.,*Patterns for e-business: A Strategy for Reuse*, IBM Press, 2001, ISBN 1931182027

Referenced Web sites

These Web sites are also relevant as further information sources:

- Red Hat home page:

  `http://www.redhat.com/`
- IBM WebSphere Commerce V5.4, Business Edition

  `http://www.ibm.com/software/webservers/commerce/wc_be/lit-tech-general.html`
- IBM WebSphere Commerce V5.4, Professional Edition

  `http://www.ibm.com/software/webservers/commerce/wc_pe/lit-tech-general.html`
- Patterns for e-business home page:

  `http://www.ibm.com/developerworks/patterns`

How to get IBM Redbooks

You can order hardcopy Redbooks, as well as view, download, or search for Redbooks at the following Web site:

ibm.com/redbooks

You can also download additional materials (code samples or diskette/CD-ROM images) from that site.

IBM Redbooks collections

Redbooks are also available on CD-ROMs. Click the CD-ROMs button on the Redbooks Web site for information about all the CD-ROMs offered, as well as updates and formats.

WebSphere Commerce V5.4 for Linux Infrastructure and

Review of e-commerce Runtime patterns and product mappings

Advanced runtime scenarios for WebSphere Commerce for Linux

Deploying a store to a Linux runtime environment

The focus of this Redpaper is to provide IT specialists with the knowledge to plan and implement a WebSphere Commerce V5.4 for Linux in a single-tier, 2-tier remote Web server or remote database, and 3-tier runtime using DB2 and the IBM HTTP Server.

Included in the implementation procedures are WebSphere Commerce FixPak V5.4.0.3 and WebSphere Application Server V4 FixPak 3. The procedures document best practices, workarounds, and FixPak procedures beyond what is documented in the original release of WebSphere Commerce V5.4.

This Redpaper is a supplement to the *WebSphere Commerce V5.4 Handbook*, SG24-6567 for the Linux platform. The appendixes include tips on Red Hat Linux and auto starting servers.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks